Retrieval and Annotation of Music Using Latent Semantic Models

Levy, Mark

Retrieval and Annotation of Music
Using Latent Semantic Models

Thesis submitted in partial fulfilment
of the requirements of the University of London
for the Degree of Doctor of Philosophy

Mark Levy

Submitted: January 2012

Centre for Digital Music
School of Electronic Engineering and Computer Science
Queen Mary, University of London

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline. I acknowledge the helpful guidance and support of my supervisor, Professor Mark Sandler.

Acknowledgements

Many thanks to all at the Centre for Digital Music, Queen Mary, without whom my research - let alone this thesis - would never have got started. Thanks in particular to all those at the Centre and elsewhere who helped me learn something about music information retrieval, and computer science in general, and have fun in the process: Samer Abdallah, Christophe Rhodes, Juan Bello, Matthew Davies, Mark Plumbley, Simon Dixon, Chris Cannam, Chris Sutton, Elias Pampalk, Michael Casey, Katy Noland, Matthias Mauch, Yves Raimond, Wen Xue, Rebecca Stewart, Emmanuel Vincent, Kurt Jacobson, Chris Harte, Andrew Robertson and many others. Above all I am grateful to Mark Sandler for giving me the opportunity to join the Centre and for his guidance and encouragement over the years.

I also owe a huge debt of gratitude to my colleagues at Last.fm: firstly for collecting much of the data that made this research possible, secondly for giving me a fascinating job, and then for allowing me time off to finish writing this thesis. Special thanks go to Norman Casagrande, Erik Frey, Klaas Bosteels and Olivier Gillet for teaching me so much, and to Phil Wilson for making my life easy.

Finally I would like to acknowledge the Engineering and Physical Sciences Research Council: research for this thesis was funded by two EPSRC grants.

If my wife Joanna Levine or any of our children Miriam, Raphael or Otto happen to be reading this then all I can say is: sorry, it won’t happen again.

Abstract

This thesis investigates the use of latent semantic models for annotation and retrieval from collections of musical audio tracks. In particular latent semantic analysis (LSA) and aspect models (or probabilistic latent semantic analysis, pLSA) are used to index words in descriptions of music drawn from hundreds of thousands of social tags. A new discrete audio feature representation is introduced to encode musical characteristics of automatically-identified regions of interest within each track, using a vocabulary of audio *muswords*. Finally a joint aspect model is developed that can learn from both tagged and untagged tracks by indexing both conventional words and muswords. This model is used as the basis of a music search system that supports query by example and by keyword, and of a simple probabilistic machine annotation system. The models are evaluated by their performance in a variety of realistic retrieval and annotation tasks, motivated by applications including playlist generation, internet radio streaming, music recommendation and catalogue search.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Writing about music is like dancing about architecture
- it's a really stupid thing to want to do

attributed variously to Elvis Costello, Frank Zappa and others[1]

1.1 Music and conventional semantics

The well-known aphorism at the top of this page expresses an intense suspicion, widely held by practising musicians, of the use of words as a way to capture anything much of value about the essential nature of a piece of music. It also echoes the formal philosophical argument that music in itself has no semantics: music is, strictly speaking, not capable of representing anything. Few of us would claim propositional semantics for music ("this piece of music is true, therefore three plus five equals seven") and amongst serious philosophers even referential semantics ("this song expresses sadness") are widely disputed [Kivy, 1997; Bicknell, 2002]. But, however much musicians and philosophers may disapprove of the practice, people do obstinately continue to write about

[1] see `http://www.pacifier.com/~ascott/they/tamildaa.htm` for a list of 17 candidate authors

music, frequently attributing it semantic properties in the process.

The research undertaken here harnesses one of the newest sources of writing about music, social tags supplied by millions of internet-savvy music lovers, in the service of an approach to automatic annotation and query-by-description inspired in part by roughly parallel work in relation to images, and often described as 'semantic'.

Although similar philosophical reservations can also apply to descriptions of images, the appeal of building systems to label photographs automatically with the names of the kinds of object that have been photographed (cat, dog, tiger, apple, river, sky, etc.) is obvious, and several models have been proposed in recent years [Mori *et al.*, 1999; Barnard & Forsyth, 2001; Jeon *et al.*, 2003; Blei & Jordan, 2003; Oliva & Torralba, 2001; Yavlinksy *et al.*, 2005]. The main hurdle to be overcome by such a system is the need to generalize not only to many different representations of a particular object, but also to a potentially unlimited number of kinds of object, which poses seemingly insurmountable issues of scalability for any deployable implementation.

Research into generating verbal descriptions of music automatically from audio recordings has been relatively limited, perhaps reflecting some caution over music's uncertain semantics. Most work in this field has cast description as a simple classification problem, either for genre, where we hope to describe music as being in one of a small number of particular widely-accepted styles, or for mood, where we want to label it as expressing a particular emotion. The main issues here are the strikingly diverse representations encountered in real music for many classes, and the inherent subjectivity of the classes themselves. These issues make it very difficult to prove convincingly that a system is performing well, and will generalize robustly. It is increasingly recognised, for example, that the timbral similarity at the heart of most state-of-the-art music classification systems captures the sound of individual artists well, but at the expense of generalizing poorly to music by artists not present in the set of tracks used to train the system [Aucouturier & Pachet, 2004; Aucouturier, 2006].

1.2 An emergent semantics of music

The social tags used as training data in this study (described in more detail in Section **??** below) provide large numbers of descriptions of individual tracks from which we can reasonably attempt to learn an *emergent semantics* of music, i.e. a set of relevant concepts which are established by a mechanism of local interaction, as individual users adopt tags which they see others are already using, rather than by global prior agreement [Aberer *et al.*, 2004]. The approach pursued here to uncover these semantics is to build models which allow us to quantify the relevance of a set of basic concepts to any given piece of music, where the set of concepts itself is learned from the data rather than imposed in advance. Using these models we can represent each track in a collection by a vector containing a relevance score for each learned concept. A particular aim in the work reported here is to build models that can learn both from tags and from low-level features extracted from the audio itself.

1.3 Aims and motivation

The primary aim pursued in this thesis is to create a semantic representation for music that can be used to generate rich descriptions of un- or sparsely-annotated tracks, and which can serve as the basis for systems for browsing and searching large music collections.

The use of a semantic representation for recordings, and of a query-by-description paradigm, is an attractive approach to searching collections of music for the following reasons:

1. query by keyword is a familiar paradigm from successful text search applications;

2. a semantic representation is well-suited to the needs of film or TV producers, etc., seeking appropriate music to match a particular mood or scenario: this is a major and growing source of revenues to the music recording industry[2];

3. a semantic representation is similarly well-suited to matching adverts to recordings, which may prove valuable as advertising-funded business models for music distribution gain in importance;

4. a (human-generated) semantic representation is already used in what is widely considered the best existing music discovery service[3].

A number of attractive internet games[4] have recently been deployed to capture human annotations for images. Their basic mechanism is to give a pair of players a short period of time in which to generate candidate labels for an image, retaining labels produced by both players. These games have proved phenomenally popular, successfully capturing annotations in vast quantities. While the underlying theoretical question ("how can we teach a machine to recognise a picture of a tiger?"), with its implications for our understanding of human vision and cognition, remains interesting in its own right, machine annotation of images for the sake of practical search applications may soon become simply unnecessary, as these games can apparently capture human annotations at a rate sufficient to support real-world applications such as Google Image Search [von Ahn & Dabbish, 2004]. Might the same be true for music? Might the growing volume of social tags equally make machine input to semantic representations of music redundant?

Although ESP-style games will no doubt become widely deployed for music, and social tagging will continue to flourish, in all likelihood their impact will be different, given the loose semantics and intrinsic subjectivity of music:

[2] see for example the June 2008 British Recorded Music Industry figures available from `http://www.bpi.co.uk`

[3] `http://www.pandora.com`

[4] The ESP Game `http://www.espgame.org`, Google Image Labeler `http://images.google.com/imagelabeler`

a reasonable expectation is that the weak semantics of music will leave a role for machine systems. In any event, the aim here is to explore the possibility of learning semantics from both words and audio. At the very least this offers a solution to the *cold start* problem, providing machine-generated descriptions for new tracks that have not yet been tagged, and which can consequently remain invisible to practical search and recommendation systems. More importantly, a primary inspiration for the research presented here was specifically the challenge of building a system that can search collections of tracks according to their sound as well as their descriptions.

1.4 A note on evaluation

How can we tell if a semantic search or annotation system for music is working well? There are some issues that we have to confront when evaluating performance on these tasks, particularly in a domain with weak semantics, and which should be borne in mind from the outset:

- basic metadata offers some sets of descriptive labels that are reasonably objective, most obviously artist identity. These categories are limited, however, and pose unrealistic classification or retrieval tasks: we are likely to represent artist identity directly in any real-world system.

- tags can serve to provide evaluation groundtruth as well as training data. This is theoretically unsound from a machine learning perspective, and in practice it also requires us to apply arbitrary thresholds to extract sets of "trustworthy" labels from tags which frequently express fallible and inconsistent opinions. Our confidence in such groundtruth will always be limited, although tags at least have the advantage of being available in high volume, reducing the impact of inconsistency in the data in comparison to the small datasets used in most previous work.

- comparative human assessment of search results and machine annota-

> tions can provide a convincing gold standard for evalution, providing we have enough evaluators to reduce the effects of subjectivity. There are practical limits, however, on the amount of such evaluation which is possible in the research environment, and there is no pre-existing expert-annotated test corpus to use as a surrogate.

The approach followed in this study is to try to draw reasonable conclusions by carrying out several kinds of evaluation in parallel.

1.5 Previous work

The formal latent semantic models described in subsequent chapters have not previously been applied to music for annotation or searching, and rarely even for other purposes (exceptions are the analysis of song lyrics in [Logan *et al.*, 2004] and the collaborative filtering system described in [Yoshii *et al.*, Feb. 2008]). Two previous systems for annotation and semantic search of music have been developed, both creating informal semantic representations from the outputs of one or more classifiers applied to audio features [Whitman, 2005; Turnbull *et al.*, 2008]. More recently an *autotagging* system, also based on a bank of classifiers, has been used to generate a wide range of artist-level annotations for use in music recommendation systems [Eck *et al.*, 2008], and similar systems for autotagging at track level are the subject of current research, notably [Mandel *et al.*, 2011a,b]. These contributions are described in detail in the following subsections, beginning with a very brief outline of the extensive literature on simple classification systems for musical audio which inspired them. Finally the following section outlines the major contributions of this thesis.

1.5.1 Simple classification systems

Research into single-label genre classification has proliferated since the seminal work of Tzanetakis and Cook in [Tzanetakis & Cook, 2002], encouraged in particular by the annual MIREX contest organised by ISMIR [Cano *et al.*, 2006; Downie *et al.*, 2005]: for recent reviews see [Scaringella *et al.*, 2006; Fu *et al.*, 2011]. Similar work has been undertaken for artist identification, and, to a much smaller extent, for mood classification [Li & Ogihara, 2003; Liu *et al.*, 2003; Wieczorkowska *et al.*, 2005]. Although reported classification accuracies for unseen test tracks are modest, unless tracks by the same artist happen to have been seen in the training set, this work has led to an extensive study of possible audio feature representations. This has included a systematic comparison of various statistics of a large number of known timbral features [Mörchen *et al.*, 2006], and even a guided brute-force method which explores a space of literally billions of features to learn the best for any particular individual classification task over a given training set [Pachet & Roy, 2007].

The main drawback of simple classification systems - even if we imagine them working perfectly - is that the output of a single label chosen from a small set of alternatives is not a very interesting description of a piece of music. This is particularly true when the music in question comes from a commercially-released recording, and when the description sought is simple categorical data such as artist or genre. In this case, the desired label is often already embedded in the audio file itself, or else can easily be looked up via a music fingerprinting service. Genre labels in particular are also frequently frustrating to music lovers, as their application can be both subjective and commercially-motivated, despite the fact that they carry high semantic significance relating to tribal notions of personal identity (Mods v. Rockers, Goths v. Punks, etc.).

Even in Tzanetakis and Cook's original paper [Tzanetakis & Cook, 2002], it was realised that the so-called *GenreGram*, a vector of classifier outputs for each competing genre class, might be a more informative representation than a single class label. As proposed, however, the GenreGram was computed for each frame of incoming audio, and used simply to provide a novel visual display to enhance music listening. A similar representation is computed in the classification system described in [West & Lamere, 2007], although the system's output is once again limited to single genre labels.

1.5.2 Whitman's bank of classifiers approach

The problem of poverty of description is turned on its head in the work of Whitman [2003; 2005], which learns a set of single-word classifiers for an automatically selected vocabulary drawn from a very large number of words found in relevant web pages. In Whitman's system, words describing artists are mined from pages on the Web and associated with a set of training tracks. Individual binary classifiers for, in principle, every single word found in the total set of pages mined, are then trained on audio features from corresponding tracks. In practice the vocabulary is thinned, for example by discarding all words apart from adjectives, but this still leaves some thousands of words each requiring their own classifier.

The apparently daunting problem of the training time required by the system is solved by the use of Regularized Least Squares (RLS) Classifiers [Rifkin *et al.*, 2003]. The RLS classifier is closely related to the well-known Support Vector Machine (SVM) [Vapnik, 1998] and uses a similar *kernel matrix* $\mathbf{K}$: $K_{ij} = K_f(x_i, x_j)$ where the kernel function $K_f(x_i, x_j)$ is a generalized dot product (in a Reproducing Kernel Hilbert Space) between training samples x_i and x_j. While training an SVM requires solving a convex quadratic program, an RLS classifier is trained simply by solving a single system of linear equations:

$$(\mathbf{K} + \lambda\mathbf{I})\mathbf{c} = \mathbf{y} \tag{1.1}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{y}$ represents the relevance of some particular word to each track in the training set. Once the inverse matrix $(\mathbf{K} + \lambda\mathbf{I})^{-1}$ has been computed, training a classifier for a new word requires only a single matrix multiplication. Once all the classifiers have been trained in this way, all but the k best-performing ones are discarded, where k is some arbitrary small number, and the output of the classifiers for the remaining few words is normalised to a vector of scores representing the relevance of each word given the audio features. The vector of scores can be used as a k-dimensional representation which Whitman shows to be a more effective input to an artist classifier than that given by the k principal components of the audio features themselves.

The weakness of the individual word classifiers, with a reported accuracy of at best around 40%, suggests that this representation is, however, more 'semantically-guided' than strictly semantic, because any individual predicted description is probably wrong. This may be an inevitable consequence of working with web-mined text, particularly when text and audio are only associated at the artist level. Web-mined text is inherently noisy, with much or even all of the text on any particular page retrieved being irrelevant to any given track which it is supposed to describe. Another unfortunate consequence of the automatic retrieval of web pages in web-mining is that the overall vocabulary size explodes, as pages from widely differing sources, each with their own characteristic vocabulary, are added to the training text. This not only adds to the issues of noise, polysemy and synonmy discussed in Section 2.5 below, but can also tie the output of Whitman's system to annotations from an idiosyncratic vocabulary.

1.5.3 Eck et al.'s boosted classifiers

Recent work on so-called *autotagging* by Eck, Bertin-Mahieux, Lamere and Green [Eck *et al.*, 2008] falls somewhere between conventional genre classification and Whitman's approach, although Eck et al. draw text training data from a large dataset of social tags for some 100,000 artists, and use more sophisticated audio feature extraction and classification methods. Like Whitman they work at the artist level, choosing to build classifiers for each of the 60 most popular artist tags according to data supplied by the Last.fm web service[5]. Over 50 of their selected tags are genre terms. They quantise the frequency with which each tag has been applied to each artist into three classes, "a lot", "some" and "none", such that equal numbers of artists fall within each class for each tag. They then

[5] http://ws.audioscrobbler.com

use a boosting algorithm [Freund & Shapire, 1996] to train classifiers for each tag on audio features from tracks by artists belonging to each class. Their audio dataset is also large, containing around 90,000 tracks in all. The accuracy of their individual classifiers ranges from 53% to 82% when their predictions are aggregated on a per-song basis, and considering the quantised classes created during the training process as a groundtruth.

The stated aim in Eck et al.'s work is to supply machine-generated tags for un- or under-tagged artists, to solve the cold start problem when adding new artists to a music recommender system, and to improve recommendations for existing artists by combining human and machine-generated tags. They give results using a range of carefully constructed evaluation measures that target these particular goals, suggesting that their machine-generated tags can indeed improved artist recommendations.

1.5.4 Turnbull & Barrington's bank of Mixture Models

Work by Turnbull and Barrington [Turnbull *et al.*, 2007a; Barrington *et al.*, 2007; Turnbull *et al.*, 2008] applies a recent image annotation system [Carneiro *et al.*, 2007] directly to music. Their initial approach to obtaining training annotations followed Whitman in mining text from record reviews [Turnbull *et al.*, 2006]. The results were poor, with at best 9% precision and 12% recall, averaged over a vocabulary of 317 words hand-picked from the total vocabulary encountered in the reviews. They attribute the weakness of these results to the difficulty of mining individual words at the track level from journalistic text. They use reviews from a single source, avoiding some of the problems of noise inherent in web-mining, but leaving a problem of semantics: the occurrence of a particular word in running text ("this track is far from *beautiful*...") does not guarantee that it names a concept relevant to the track under review.

They now deal with the issues of both noisy data and uncertain semantics by paying university students to provide training descriptions in the form of questionnaire answers. Each question addresses the relevance of one of a hand-picked set of 135 concepts to the track being assessed, and each track is annotated by at least three students, who are required to supply answers about all of the concepts for each track they annotate. The answers are used to estimate the relevance of each concept to 500 training tracks, each by a different artist. A Gaussian Mixture Model for each concept is then trained on features for each track for which the concept was assessed as relevant, with some tying of parameters to reduce computational cost. A simple weighting scheme is used during training to reflect the degree of relevance of the concept to each training track. Relevance is based on the proportion of students who annotated a particular track with the concept, penalised by the proportion who marked it as irrelevant. When presented with a new track, the system labels it with the k most likely concepts according to the GMMs, where k is some arbitrary fixed small number. Although in absolute terms the results are modest, the system does no worse than similar image annotation systems, with annotation precision of 27% and recall of 16% averaged over the concepts, when $k = 10$ machine labels are output for each track. To create a groundtruth for this evalution, Turnbull and Barrington apply a threshold to the relevance scores used in training: concepts with relevance scores above this threshold are considered "reliable" annotations.

Interestingly the performance of a *human baseline*, created by holding out a single questionnaire answer for each concept, is poor, doing no better than the system itself. This suggests that a large number of questionnaire respondents, and a robust method to combine their often conflicting responses into a single relevance score, would be required before evaluation against such data could be considered reliable. No doubt aware of this issue, in ongoing work Turnbull and Barrington are pursuing more scalable data collection through an ESP-style game [Turnbull *et al.*, 2007b].

1.5.5 Mandel et al's Restricted Boltzmann Machines

Recent efforts both to improve data collection for semantic annotation, and to extend the artist autotagging approach of [Eck *et al.*, 2008] to individual tracks using a variety of different classifiers, are reported in a series of papers by Mandel. Mandel's own ESP-style *MajorMiner* online game, in which players score points for tagging ten second clips, is described in [Mandel & Ellis, 2007]. Players can tag clips at any time, but points are assigned for a tag only when it is validated by a second player. A second approach, using Amazon's Mechanical Turk system[6] to employ unknown workers to provide tags, is explained in [Mandel *et al.*, 2010]. In this study multiple short clips from the same song are offered for annotation, the aim being to collect data allowing investigation of how descriptions might change over the course of a song. Although these methods might be scalable in principle, the actual datasets collected in practice appear to be rather modest: the two papers report tags for single clips from 2000 songs, and multiple clips from 185 songs, respectively. Mandel also reports results on a larger collection of social tags for 9000 tracks drawn from a dataset originally collected for a study on friendship in social networks [Schifanella *et al.*, 2010].

These three datasets are used to evaluate a classification-based autotagging system in experiments exploring various choices of classifier and low-level audio feature representation [Bergstra *et al.*, 2010; Mandel *et al.*, 2010, 2011a,b]. A key feature of recent versions of Mandel's system is the use of a restricted Boltzmann machine (RBM) [Smolensky, 1986] in a preprocessing step designed to improve the training data used for classification; an RBM is also evaluted as a possible implementation of the classification layer itself.

An RBM is a generative probabilistic model implemented as a two-layer stochastic neural network of binary units i.e. each unit can be on or off. One layer contains *visible* units corresponding to observed values, while the other contains *hidden* units. The network topology is *restricted* in an RBM in the sense

[6] `https://www.mturk.com`

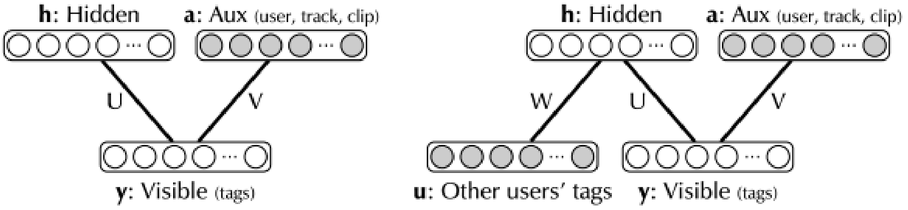


Figure 1.1: Mandel's restricted Boltzmann machines

that units in each layer are connected to every unit in the other layer, but not to one another. In practice the RBM is designed so that the layer of visible units corresponds to a binary feature vector. Training then consists of optimizing the parameters by gradient descent to minimise the log-likelihood of the data i.e. of the states of the network where the visible units take values seen in the training features. This can be accomplished efficiently by an approximate sampling method known as *Contrastive Divergence* [Hinton, 2010].

The units in an RBM can also be conditioned on auxiliary variables. The networks used in Mandel's pre-processing step are illustrated in Figure 1.1, reproduced from [Mandel *et al.*, 2011b]. During training the visible units $\mathbf{y}$ are set to the tags applied to a clip by a specific user, while the additional units in layer $\mathbf{a}$ represent user and clip identity, and those in layer $\mathbf{u}$ the tags applied to the same clip by other users. After training it is possible to "smooth" the tags applied to a clip by replacing raw observations $\mathbf{y}$ with $p(\mathbf{y}|\mathbf{a})$ or $p(\mathbf{y}|\mathbf{a},\mathbf{u})$, as estimated by the model. This smoothing corresponds to introducing unseen tags that have been applied to other clips in the same track, or that have been frequently applied together with the observed tags.

Smoothed tag associations output from this pre-processing step are then used to train either a bank of binary classifiers, one per tag for a fixed vocabulary of common tags, or a single further *discriminative* RBM [Larochelle & Bengio, 2008] designed to function as a multi-label classifier. Performance is reported on a per-tag basis, with evaluation done against the raw tag associations. Both the pre-processing step and the use of a single multi-label classifier are shown to improve classification accuracies, although using both refinements together does not improve results further, presumably because they learn the same dependencies between tags. The overall performance remains fairly weak, however, with on average slightly more than two of the top ten tracks predicted to have any particular tag actually having that tag in the groundtruth [Mandel *et al.*, 2011b].

1.5.6 The MIREX tag classification contest

Data collected by the online game described in [Mandel & Ellis, 2007] has also been used as the basis of an Audio Tag Classification task in recent rounds of the annual MIREX algorithm contest organised by ISMIR, devised to support research into autotagging unlabelled audio [Bertin-Mahieux *et al.*, 2010]. The contest dataset has a vocabulary of 43 tags, and contains clips from 1400 tracks. Algorithms are required to perform a binary classification task for each tag, where the test set contains equal numbers of clips that have and have not been assigned the tag in question. They must also rank tracks for each tag, and rank predicted tags for each track. The rankings are evaluated using a binary relevance criterion, but with metrics that reward algorithms for predicting correct associations ahead of incorrect ones. Groundtruth relevance itself is established by simple "verification" i.e. a tag is regarded as correct if it has been applied to a clip by more than one player of the game.

Some progress has been made against these metrics, with the current state of the art reflected by [Hamel *et al.*, 2011]. This uses a Multi-Layer Perceptron [Rumelhart *et al.*, 1986] as a multi-class classifier, trained on various simple statistics of audio features. The MLP is a neural network with a single hidden layer, making it similar to the discriminative RBM of [Mandel *et al.*, 2011a], although many details of feature summarisation and training are different. In particular these models appear to do well because the parameters of hidden layers can be learned by pretraining on large quantities of unlabelled audio data. In practice pretraining can be done ahead of the contest, compensating for the small size of the contest dataset, which in itself may offer too little training data to support significant machine learning [Marques *et al.*, 2011].

The contest also evaluates algorithms on a second dataset of 3,500 tracks labelled as belonging to one or more of 18 mood groups, where the mood labels have been established by human moderation of social tags [Hu *et al.*, 2009]. Over half the songs in the dataset are marked as belonging to more than one mood group, going some way to addressing the shortcomings of the simple classification tasks referrred to in Section 1.5.1. Relative algorithm performance is consistent between the two datasets, although absolute performance is slightly worse when predicting mood labels.

1.6 Modelling semantic relevance

An obvious theoretical shortcoming of most existing work on semantic annotation of music is the use of a classification approach in the absence of a principled model for semantic relevance: instead ad hoc thresholds have to be used to decide whether or not a particular word is relevant to any given track. This leads to problems of data sparsity, potential misclassification during training and evaluation, and difficulty in handling weak but significant associations between words and tracks.

Although Whitman uses sophisticated techniques for text mining, his classification system implicitly uses a naive binary semantic relevance model during training and evaluation: each word in the vocabulary is regarded either as relevant to a given track (if the word occurs anywhere in its artist's associated training text) or irrelevant (if it does not). Turnbull & Barrington's questionnaire also asks listeners simply to rate concepts as either relevant or irrelevant. Of course listeners do not always agree, and so Turnbull & Barrington use a straightforward ad hoc model to aggregate conflicting answers into a continuous relevance score. Despite their impressive dataset of millions of individual tags, and their decision to annotate artists rather than individual tracks, Eck et al. still complain of a problem of data sparsity in relation to their relatively modest target vocabulary of sixty terms. Mandel's approach appears to capture a more refined measure of semantic relevance, although this remains hidden in the parameters of his ingenious "smoothing" model, and his work remains focussed on binary classification.

Simple relevance models like these clearly look to be dangerous given the weak semantics of music. Conflicting opinions amongst listeners, and weak associations between text and the music it purports to describe, are the norm. This poses particular problems for machine annotation. It is difficult, for example, to choose positive training examples for any given concept with confidence, to create a trustworthy human 'groundtruth' for evaluation, or to avoid false negatives during evaluation, e.g. when the machine outputs *strings* for a track annotated only as *violin*. What looks to be needed is a better way of measuring the relevance of concepts to a given track: in the work presented here, well-understood latent semantic models are used to address this.

1.7 Contribution of this thesis

The remainder of this thesis develops and evaluates an information retrieval approach to semantic search and machine annotation of music, using data from both social tags and audio content, as an alternative to the classification-based methods described in the preceding Sections. Also in contrast to most previous work, the methods developed in this study work at the track rather than the artist level, in order to model more directly the relationship between sound and description.

The work is organised broadly as follows: Chapter 2 shows how retrieval models can be applied to data from social tags; Chapter 3 proposes a discrete method of modelling audio features to make them easily compatible with tag data; Chapter 4 extends the models to the resulting joint vocabulary; Chapter 5 shows how these models can be used for real-world tasks such as annotating sparsely-tagged tracks and supporting query by description search. Finally

Chapter 6 focusses specifically on modelling emotion words in tags, leading both to proposals for novel interfaces for browsing and searching large music collections, and, less expectedly, to results of interest to the study of the psychology of music.

The following Sections give a more detailed overview:

Chapter 2

This Chapter begins by motivating the use of ranked retrieval methods in preference to a classification approach for practical applications in the domain of music. It then gives an introduction to social tags in general, and social tags for music in particular. A set of some 660,000 tags for tracks is collected, and certain characteristics of tags for music are identified: these characteristics inform the choice of how best to interpret the tags we see applied to tracks as data expressing semantic relevance. The resulting data is first modelled in a simple vector space. Latent semantic models, derived respectively by geometric and probabilistic methods, are then introduced. An experimental framework is established, together with standard evaluation metrics, allowing reasonable comparison with previous work. A set of retrieval tasks is then used to compare the models both with one another and with a baseline from the literature. Special attention is paid to the extent to which the models are able to generalise to tracks by artists for whom there were no tags in the training data. The results show that a vector space model based on tags outperforms previous methods in the literature on simple artist identity and genre retrieval tasks, while pobabilistic latent semantic models in particular show encouraging ability to generalise to unseen artists. Finally the specific semantic aspects learned by the latent semantic models are illustrated and discussed.

Chapter 3

Having established the effectiveness of semantic models based on tags in Chapter 2, Chapter 3 explores an approach to extending them to model audio content. The aim here is to develop a representation of low-level audio features as a vocabulary of discrete audio *muswords*, where each musword can be associated more or less strongly with any particular track in exactly the same way as a conventional word. The models can then be extended easily to a joint vocabulary of words and muswords. The Chapter begins by expanding the motivation for incorporating audio information in our models despite the excellent performance of models based purely on tags reported in Chapter 2. It then introduces a method of selecting timbral and rhythmic features for automatically-identified regions of interest within a given track, and proposes two alternative ways in which the resulting features can be mapped onto a vocabulary of muswords. The experimental framework of the preceding Chapter is reused to evaluate the performance of muswords on retrieval tasks using a simple vector space model: the features and discretisation methods are compared with one another and with related work in the literature. The best performing musword representation is found to outperform previous discrete methods in the literature, being comparable with state of the art methods based on a raw audio feature representation, paving the way for effective joint semantic models.

Chapter 4

This Chapter shows how the semantic models of Chapter 2 can be applied effectively to the joint vocabulary of words and audio muswords established in Chapter 3. In general, given the state of the art in low-level audio feature representations, the semantic information contained is words is far more reliable than that offered by audio content. This Chapter therefore looks in detail at how words and muswords can be combined effectively, particularly in cases where tags are sparse and audio information has to be relied upon: such cases are all too common in real music collections. This leads to the development of an experimental framework in which retrieval tasks are evaluated under increasing conditions of tag sparsity. Combining words and muswords in a single vocabulary raises the issue of how to scale the association between John Coltrane's *Giant Steps* and audio feature *zQy432*, say, relative to the association between that track and the word *jazz*. The framework is used to study this scaling and also to compare two different training methods for a joint aspect model; again all results are compared with a baseline. Given a suitable training method, the inclusion of audio muswords in a joint model is found to improve retrieval performance for sparsely-tagged tracks with no loss of performance for well-tagged ones.

Chapter 5

In this Chapter the models developed in Chapters 2-4 are finally applied to the problems that originally motivated them: automatic annotation of sparsely or un-annotated tracks, and semantic retrieval. A realistic experimental framework is established, simulating the current real-world availability of tags, so that, for example, 30% of the tracks in the test set have no annotations at all. Performance on these tasks is then evaluated for a range of latent aspect models trained jointly on tags and audio. Annotation performance is found to be equivalent to a comparable classification approach, while retrieval performance is roughly twice as good as that of the most similar system reported in the literature.

Chapter 6

This Chapter studies emotion words in social tags for music. It relates the low-dimensional latent semantic spaces learned by the models of Chapters 2-5 to the emotion spaces studied for decades by music psychologists, and shows how data from tags can be used to update and extend the psychological models. Semantic models are also used to investigate the correlation between emotion words and musical genre. Finally the close relationship of latent semantic and psychological models is used as a basis to propose novel interfaces for browsing and searching large collections of music.

Chapter 7

The final Chapter gives an overview of the work presented here, with a critical analysis of its strengths and weaknesses, and outlining proposals for further work.

1.7.1 Major contributions

The major contributions of the thesis are in the following areas:

1. this study introduces the use of text information retrieval methods to analyse social tags for music
2. latent semantic models are applied to capture the relevance of individual words to tracks from a large dataset of tags
3. a new discrete audio feature representation is introduced, based on automatically identified regions of interest within each track, enabling the extension of these models to audio information
4. a simple probabilistic setting for machine annotation is proposed in place of a classification approach and evaluated
5. a joint aspect model, able to learn descriptions of music from both tagged and untagged tracks, is developed and evaluated

While this work naturally uses semantic models to represent tracks by the set of descriptions applied to them, the symmetry of the models also allows us to represent descriptive words by the tracks to which they are applied. This approach is pursued here in a study of emotion words applied to music in tags, demonstrating the potential broader application of semantic models based on tags as computational tools within music psychology and musicology. The major contributions here can be summarised as follows:

1. semantic models of tags are introduced to the study of emotion in music, suggesting changes to traditional models of affect
2. a novel psychologically-motivated user interface to large collections of tracks is proposed, based on analysis of the co-occurrence of tracks and emotion words in tags

1.8 Publications

Parts of the research reported in this thesis were previously published in the following journal and conference papers[7]:

- Journal Papers

  Levy, M., & Sandler, M. 2008a. Structural segmentation of musical audio by constrained clustering. *IEEE Trans. Audio, Speech and Language Processing*, **16**(2), 318–326. [Levy & Sandler, 2008a] (citation count 54)

  Levy, M., & Sandler, M. B. 2008b. Learning latent semantic models for music from social tags. *Journal of New Music Research*, **37**(2), 137–150. [Levy & Sandler, 2008b] (citation count 22)

  Levy, M., & Sandler, M. B. 2009. Music Information Retrieval Using Social Tags and Audio. *IEEE Trans. Multimedia*, **11**(3), 383–395. [Levy & Sandler, 2009] (citation count 31)

- Conference Papers

  Levy, M., & Sandler, M. 2006a. Lightweight measures for timbral similarity of musical audio. *In: Proc. 1st ACM Workshop on Audio and Music Computing for Multimedia*. [Levy & Sandler, 2006b] (citation count 20)

[7] citation counts retrieved from Google Scholar on 17 January, 2012

Levy, Mark, Sandler, Mark, & Casey, Michael. 2006. Extraction of high-level musical structure from audio data and its application to thumbnail generation. *In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. [Levy *et al.*, 2006] (citation count 42)

Levy, M., & Sandler, M. 2007. A semantic space for music derived from social tags. *In: Proc. 8th International Society for Music Information Retrieval Conference*. [Levy & Sandler, 2007] (citation count 59)

Stewart, Rebecca, Levy, Mark, & Sandler, Mark. 2008. 3D interactive environment for music collection navigation. *In: Proc. 8th International Conference on Digital Audio Effects*. [Stewart *et al.*, 2008] (citation count 10)

Chapter 2

Learning semantic models for music from social tags

In the age of physical recordings of music (LPs, singles, cassettes, CDs, etc.) the primary form of organisation for collections of recorded music was the recording catalogue. The system embodied in these catalogues, whether in their original form as printed books or in more recent digital incarnations, was essentially a tree structure encompassing basic metadata, with nodes representing record labels or genres near the top of the tree, nodes below these representing artists, and leaves representing albums or other releases. This organisation can still be seen in the fully digital era in the design of personal media collection managers such as iTunes, and in the menu structures of some online music retailers. While this may be a symptom of slow innovation in the music business, or of intellectual conservatism on the part of music consumers, it also suggests that there is some intrinsic value in the genre-artist-album tree structure as a mechanism for organising music collections.

The history of internet search, however, suggests strongly that such rigid tree structures rapidly prove inadequate as the number of digital documents available grows. More flexible and powerful systems for searching and browsing digital music collections (let alone all music available on the web) are likely to require models similar to those familiar from text information retrieval. In these models the similarity between tracks, or between tracks and arbitrary text queries such as "laid back piano jazz", can be expressed either in terms of conditional probabilities or as distances between points in some vector space. This Chapter shows how existing methods from the field of text search can be used to learn latent semantic models for music, in which individual tracks are represented as points in relatively low-dimensional spaces. The spaces are computed by geometric or probabilistic dimension reduction from very high-dimensional term vectors, representing the relevance to each track of a large vocabulary of words found in social tags. The learned dimensions are considered *semantic* because they appear to capture a set of significant underlying concepts for the collection of tracks being modelled.

The models described in this Chapter can be used to support various kinds of information retrieval on collections of tracks, including nearest neighbour search and query by keyword, and their performance can be measured in many different ways. In this Chapter we focus on evaluating the extent to which nearest neighbour search with these models respects a traditional recording catalogue organisation, in which tracks are grouped by artist and genre. A good deal of work has been devoted to addressing this issue in relation to models based on low-level audio features in the hope of building playlist generation and music recommendation systems based on audio analysis (see Scaringella *et al.* [2006] for a recent review). The conclusion, after several years of research, is that current low-level feature sets lead to a representation that is only weakly structured by artist and genre Aucouturier [2006]; Pampalk [2006]. While there is no 'right answer' for the degree of organisation that is required for practical search systems, because, for example, songs by other artists or from different genres can quite reasonably be considered very similar to any given query track, organisation by artist and genre is well understood by music lovers, and the lack of such organisation in low-level feature representations appears to be a major barrier to their acceptance in practical applications.

One possible reason for the poor performance of existing audio content-based models is that the data representation, with tracks or sections of audio summarised typically with the order of 100 feature statistics or model parameters, is simply not rich enough to capture the complexity of music in general. We investigate the performance of our models as the number of latent dimensions varies to see if low-dimensional semantic representations improve or degrade our ability to search music collections, and whether there appears to be a 'natural' dimensionality to the space of descriptions of music collections, and we measure the extent to which a semantic model trained on a particular collection can generalise to unseen tracks. Finally, where possible, we also attempt to understand which concepts are expressed by the learned dimensions of our models.

The rest of this Chapter is organised as follows: in Sections 2.1, 2.2 and 2.3 we motivate the use of social tags as our underlying source of data and discuss the particular nature of social tags for music; in Section 2.4 we describe a basic vector space model for tracks based on tags; in Sections 2.5 and 2.6 we introduce latent semantic models derived respectively by geometric and probabilistic methods; in Section 2.7 we describe our experimental framework and evaluation metrics, giving results in Section 2.8 and 2.9, and drawing conclusions in Section 2.10.

Some of the work in this Chapter was previously published in [Levy & Sandler, 2007, 2008b].

Table 2.1: Symbols used in this thesis

symbol	**meaning**
t, t_i	a track, the i-th track in a collection
$\mathcal{T}$	a collection of tracks $\{t_1, t_2, ..., t_N\}$
N	number of tracks in a collection
g	a tag, may contain several words
G	number of distinct tags applied to a collection
G_w	set of tags containing word w
$f(t, g)$	number of times tag g has been applied to track t
F	total number of tags applied to a collection
w, w_j	a word or term, the j-th word in the vocabulary
$\mathcal{W}$	vocabulary of words $\{w_1, w_2, ..., w_M\}$
M	vocabulary size i.e. number of distinct words applied to a collection
$n(t, w)$	number of times word w has been applied to track t
$n(t)$	$\sum_w n(t, w)$ i.e. total number of words in tags applied to track t
$\mathbf{N}$	track-term matrix of counts $n(t, w)$
$N(w)$	number of tracks tagged with word w
R	number of track-word pairs where $n(t, w) > 0$
z, z_k	a latent aspect, the k-th latent aspect of a model
K	number of latent aspects
m	a musword
y_m	features for musword m
$c(t, m)$	count for musword m for track t
D	musword vocabulary size

Notation

For consistency, in this and subsequent and Chapters the notation given in Table 2.1 is used wherever possible when referring to the basic concepts of interest here. These concepts are also used as far as possible when introducing models and algorithms from conventional text Information Retrieval, so for example we talk about 'tracks' rather than 'documents'. Note that 'word' and 'term' are used interchangeably, while 'tag' refers to the complete annotation attached to a track by a listener, which may consist of more than one word.

2.1 Social tags for music

Social or collaborative *tags* are brief descriptions supplied by a community of internet users to allow navigation through large collections of uncatalogued media [Wu *et al.*, 2006; Golder & Huberman, 2006]. Tags therefore aid browsing or searching types of material which are not yet well served by fully automatic information retrieval techniques. Some well-known tagging systems are those offered by flickr (digital photographs), Technorati (blog posts), or del.icio.us (favourite web links)[1]. Users might tag an image on flickr, for example, as "beach", "vacation", "summer", "santa barbara", "blue sky", etc. Such tags are described as social because they are automatically shared with all other users. This implicit collaboration makes it possible to annotate large collections of documents so that they become navigable by keyword: "find me all pictures tagged beach and blue sky", etc.

As well as overcoming issues of scale in annotating large collections, the sharing of tags encourages the emergence of a common tagging vocabulary. Although tagging conventionally places no restrictions on the text that can be used as a tag, the shared purpose of creating a usable navigation system makes it attractive for users to select tags which others are already using. New tags consequently enter the mainstream vocabulary in an "organic" fashion as they become adopted by significant numbers of users. This can lead to the development of *folksonomies*, entire taxonomies reflecting current usage amongst the user community, offering a different view to the traditional categories of library cataloguing.

2.2 The nature of tags for music

With this model in mind, we might expect users to take advantage of tagging systems for music by tagging tracks directly with a vocabulary of relevant concepts, presumably encompassing things such as mood and function in addition to genre and music-specific technical terms, expanding the tree of basic metadata into a rich folksonomy of significant terms. This is arguably the 'classical' tagging scenario, embodied in most current tag-based search interfaces, which

[1] `http://www.flickr.com`, `http://www.technorati.com`, `http://www.delicious.com`

highlight the most widely-used tags for the page or item in question, and offer a naive search facility based on direct matching of tags. It also forms the basis for the *collaborative structured tagging* intended to power new knowledge-sharing ventures such as Amazon's amapedia[2]. This model, however, appears unrealistic for social tags for music.

Basic catalogue information (artist, track title, etc.) is already available for most recordings in embedded ID3 tags, or can be found by a straightforward request to a look-up service such as CDDB or MusicBrainz[3]. This information is therefore automatically made available to listeners by standard media players such as iTunes or Windows Media Player. Tagging, however, can support navigation through large collections of music according to categories which are more relevant to the role of music in everyday life: "find me music that's good to exercise to", etc. Tags also allow the expression of personal or "tribal" responses to particular songs or performers which are central to the characteristic use of music to define one's social identity. Because the vocabulary of tags is unconstrained, this self-expression can go far beyond a simple advertisement of the user's musical likes and dislikes, potentially allowing users to share and compare their emotional responses to, or categorisations of, music freely with millions of peers.

Data collection

To investigate further, a data set of 667,900 tags for 31,359 individual tracks by 5,265 artists was aggregated from two of the most important sources of tags for music. The tags were downloaded from the Last.fm[4] and MyStrands[5] web services between March and August 2007. Information about the individual users who applied the tags is not available, but the Last.fm service provides 'counts' indicating the relative number of times each tag has been applied to

[2] http://amapedia.amazon.com/view/Meta:About

[3] http://www.gracenote.com, http://musicbrainz.org

[4] http://ws.audioscrobbler.com

[5] https://www.musicstrands.com

the track in question. The MyStrands service lists all the tags ever applied to each track, while the Last.fm service returns a maximum of the 100 most frequently applied tags for each track. Although the terms of use do not permit redistribution of the data set as a whole, it was acquired through standard documented calls to the public Last.fm and MyStrands web service APIs, and a similar data set can be obtained as long those APIs remain available.

Simple statistics of these tags as shown in Figures 2.1 and 2.2 demonstrate that they are far from constituting a vocabulary of basic concepts, even allowing for a large amount of error, subjectivity or other statistical noise. In the first place, tags for music are often discursive, as illustrated in Fig. 2.1, which shows the number of tags in our data set against their length in words. We can observe that over a third of the tags consist of three or more words, while over 10% contain five or more words: these are frequently complete phrases. Secondly, the vocabulary of tags shows no sign of converging to a taxonomy as the number of tags grows. Rather the vocabulary grows according to the power law, known as Heaps' Law, characteristic of ordinary text documents, as shown in the log-log plot of Fig. 2.2. Heaps' Law is given by

$$G = \kappa F^b \tag{2.1}$$

where G is the number of distinct tags and F is the total number of tags applied, and κ and b are constants for the given collection of tracks. The vocabulary growth which we observe for tags for music fits very closely to $b = 0.42$ once we consider a large number of tags, in line with typical values seen in standard text corpora [Manning *et al.*, 2008]. Table 2.2 shows the first few tags we downloaded containing the term *80s*, illustrating the freedom with which words are combined even in short tags.

There are various reasons why we might expect to see a large vocabulary of tags applied to music. In the first place, music's weak semantics compared to other media can make the selection of tags highly subjective. There may be few

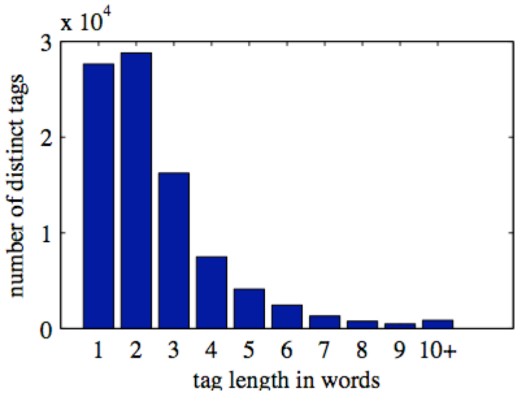


Figure 2.1: Tag lengths

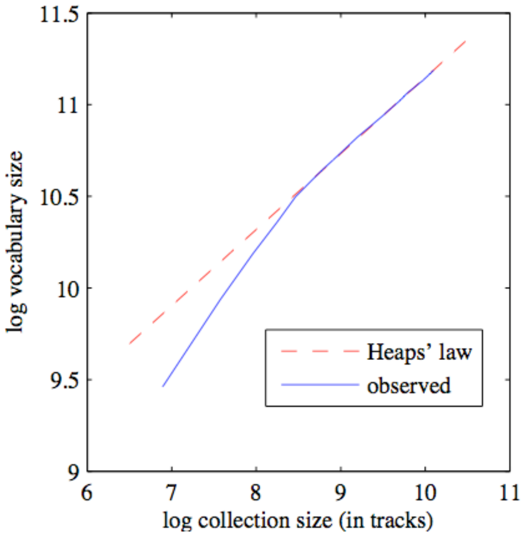


Figure 2.2: Tag vocabulary growth obeys Heap's law

Table 2.2: Some tags containing the term *80s*

80s
80s rock
My 80s memories
80s y 90s
80s and 90s
60s 70s 80s rock
80s and 90s rnb
80s wave
80s-90s
80s Music
flya 80s
Decade: 80s
80s Classic
we love the 80s
80s magic
big-hair 80s
20 songs mix : 80s Hits
golden 80s
80s alternative
ilx 80s poll
The 80s was not a dead decade
pop 80s
80s soundtracks
80s Pop
80s throwback
80s songs i love

'obvious' choices of tags for a new track, compared, say, to choosing tags for a photo of a dog, or for a blog about a particular new operating system. To judge from available figures (the ones we could find are not as recent as our data for music) the vocabulary of tags for music does indeed grow faster than that for photos or urls. A dataset of tags for 9 million Flickr photos had a vocabulary of over 200,000 terms [Schmitz, 2006], and a comparable set of Del.icio.us tags for 690,482 urls had 126,304 distinct tags [Wu *et al.*, 2006]: the statistics of our tags would predict over 800,000 and 300,000 distinct tags for this number of photos and urls respectively.

The language of tags for music is ad hoc and highly informal, suggesting that tags are frequently supplied in a spontaneous and unreflecting manner, and may say as much about the tagger as about the piece being tagged, as shown by the following selection of tags chosen at random: 'all my hope is gone', 'oregon trips', 'my favourite muse songs', 'french-canadian', 'Tool Mix', 'comp1', 'ragga rhythm', 'Dave Brubeck Quartet', 'american wedding', 'fora do mundo', 'space trucking', 'right in two', 'desert island songs - songs which keep me alive or otherwise enrich entertain and edify - the best songs in the world', 'heard on 96wave', 'put on mikey cds'. Longer tags can verge on the poetic: 'good for dancing to in a goth bar if you can muster sufficient abandon and like getting the evil eye', 'if you fall in love with me you should know these songs by heart', 'sure go ahead and depress the hell outta me what do i care', supporting the view that tags for music should primarily be regarded as a form of free self-expression on the part of the tagger. Tags most certainly do provide a novel source of information about personal responses to music, which we can bring to bear, for example, on various classic questions in the study of music psychology: this approach is pursued further in Chapter 6.

In this and the following Chapters, however, it is assumed simply that, despite the vagaries of individual tags, patterns of co-occurrences of words in tags can reveal terms or combinations of terms which are

1. significantly grounded in the music they describe, rather than expressing

arbitrary personal reactions; and

2. generalisable across tracks.

The set of tags for each track is consequently treated as a Bag of Words (BOW), following the standard information retrieval approach to text documents. Tags for each track are first tokenized with a standard stop-list to remove extremely common words with little or no semantic content ('it', 'and', 'the', etc). The tags are then tabulated as entries in a *track-term* matrix $\mathbf{N}$ of co-occurrence counts $n(t, w)$ similar to the *document-term* matrix familiar from conventional text IR, where $n(t, w)$ represents the number of times we see the word w in tags applied to track t. In contrast to standard practice with traditional text documents, a stemmer is not used to strip word endings (so that, for example, "singer", "sings" and "singing" would all be recorded simply as "sing"). Although in principle it might be advantageous to use a stemmer, existing algorithms can be expected to fail in many cases due both to the idiosyncratic vocabulary of social tagging, and to the large number of standard dictionary words used as proper nouns, particularly in artist names used in tags (for exiample we would most likely not want to stem "talking heads" or "rolling stones"). Using words rather than entire tags as the basic unit of data nonetheless goes some way towards capturing the common meaning of alternate forms such as 'female vocalist', 'female vocals', 'good female vocals', 'sexy female vocals', 'lovely female vocals', etc.

In practice only partial information is available about the number of times that each tag has been applied to a given track. The Last.fm web service gives integer percentages relative to the most frequently applied tag, with the frequency of relatively rare tags rounded down to zero: this enforces a form of editorial censorship in the tag clouds shown on web pages, where by convention the font size for each tag is proportional to its count, i.e. tags with zero counts are simply not displayed. The MyStrands web service gives only a list of tags applied to each track, with no information about their relative popularity. Following some initial experiments (reported in full in [Levy & Sandler, 2007]), it was decided to discard the MyStrands data, and to expose our models to all the Last.fm tags, including those with zero counts, simply by incrementing the published numbers. Formally we set

$$n(t, w) = \sum_{\mathrm{G}_w} f(t, g) + 1 \tag{2.2}$$

where G_w is the set of tags containing the word w, and $f(t, g)$ is the frequency with which tag g has been applied to track t according to the Last.fm web service. This resulted in an overall matrix of roughly 25,000 rows (tracks) and 30,000 columns (words). In our experiments the data was naturally split into various training and test sets: exact details of the size of these datasets are given later in Table 2.4.

2.3 Tags vs web-mined text

This approach to tags makes them directly comparable with the web-mined text, particularly blogs and music reviews, used in various academic studies as a source of high-volume descriptive metadata for music [Baumann & Hummel, 2003; Whitman, 2005; Knees *et al.*, 2004]. Although some interesting preliminary results have been reported, two significant problems are associated with mining descriptive metadata from the web. Firstly, the text retrieved is usually noisy, i.e. it unavoidably contains a great deal of irrelevant content. Secondly, for computational reasons, and because the noise problem becomes insuperable, text has to be mined on a per-artist rather than per-track basis. Social tags as applied to individual tracks appear to offer a solution to both of these issues.

Web-mined text is typically retrieved by searching for pages that appear to be relevant to a particular artist, and then attempting to retain only terms that relate to their music [Baumann & Hummel, 2003; Whitman, 2005]. The resulting text is inherently noisy on two levels. Firstly, the pages retrieved by

Table 2.3: Top terms describing Portishead

Tags	**Web-mined text**
trip-hop	cynical
electronic	produced
portishead	smooth
female vocalists	dark
downtempo	particular
alternative	loud
mellow	amazing
chillout	vocal
sad	unique
90s	simple

any automated system are not guaranteed to be relevant (in particular when an artist's name has other meanings), and come from a variety of kinds of source, each with its own characteristic vocabulary. Secondly, in general only a small unknown part of the content of each page will refer directly to music of interest. One consequence of the inevitable inclusion of irrelevant terms is that the vocabulary size explodes. A typical web crawl reported in [Knees *et al.*, 2004] found over 200,000 terms for a set of 200 well-known artists. In contrast, we found less than 13,500 distinct tags for tracks by the same set of artists. Such a comparison is necessarily informal, because of the difficulty of comparing the sizes of the input data sources (50 web pages vs tags from many different users for each artist). More importantly, however, web-mining appears to be impractical as a source of metadata at the track level, as the problems of noise multiply still further.

The vocabulary of tags is different from web-mined text not only in size, but also in character, as illustrated in Table 2.3, which compares the ten most widely applied tags in our dataset for the group Portishead with the top web-mined adjectives given in [Whitman, 2003]. We observe that, in contrast to the tags, as many as half of the web-mined adjectives ('cynical', 'produced', 'particular', 'amazing', 'unique') are very unlikely to be grounded in the music of this particular group.

2.4 Vector space model

In the well-known Vector Space Model for information retrieval [Salton *et al.*, 1975], a weighting scheme is applied to the entries of the track-term matrix $\mathbf{N}$, and a distance measure between vectors of weighted counts $\hat{n}(t, w)$ is chosen as the matching function between tracks and queries. Queries can be either free combinations of words or, in the query by example scenario characteristic of music applications such as playlist generation and recommendation, tracks themselves, represented by their term vectors, i.e. their entire vectors of weighted word counts.

For our baseline model we use the standard tf-idf (term frequency - inverse document frequency) weighting

$$\hat{n}(t, w) = n(t, w) log \frac{N}{N(w)} \tag{2.3}$$

where N is the total number of tracks and $N(w)$ is the number of tracks tagged with word w. To compare queries and tracks we use a standard matching function, cosine distance

$$s(t, q) = \frac{\sum_w \hat{n}(t, w)\hat{n}(q, w)}{\sqrt{\sum_w \hat{n}(t, w)^2}\sqrt{\sum_w \hat{n}(q, w)^2}} \tag{2.4}$$

While the implementation details vary in practice, the basic algorithm for retrieval using a vector space model to find the top r tracks matching a query q in a collection of N tracks is given in Algorithm 2.1 [Manning *et al.*, 2008, section 6.3]. The so-called document $Length$ for track t is simply the normalising term $\sqrt{\sum_w \hat{n}(t, w)^2}$ from the denominator of Equation 2.4; in practice the Lengths for tracks in the collection will be computed in advance of query time. Note that the Length of the query is neglected in Algorithm 2.1 because it depends only on the query and so does not affect the ranking of the tracks in the collection.

Algorithm 2.1: Basic vector space retrieval algorithm

```
Input: Query $q$, number $r$ of desired hits
Output: Top $r$ best matching tracks
Scores[$N$] ⟵ 0;
Compute Lengths[$N$];
foreach Query Term $w$ in $q$ do
    Calculate $\hat{n}(q, w)$;
    Fetch list of tracks containing $w$;
    foreach Track $t$ in list do
        Scores[$t$] ⟵ Scores[$t$] + $\hat{n}(t, w) * \hat{n}(q, w)$;
    end
end
foreach Track $t$ do
    Scores[$t$] ⟵ Scores[$t$] / Length[$t$];
end
Return tracks with the highest $r$ Scores;
```

2.5 Latent semantic analysis

Retrieval in the Vector Space Model depends on exact matches between the words present in queries and documents, and is therefore subject to problems of *polysemy* (where the same word is present in two documents but with different meanings), *synonymy* (where different words are present but with identical meanings), *noise* (where matching but irrelevant words are present), and *data sparsity* (where a relevant word is not present). Synonmy and data sparsity are a common problem when modelling social tags for music, because, for example, we want the query 'electronica' to retrieve tracks that have been tagged 'electro', 'electronic', etc., and we cannot guarantee that all popular variants will have been applied to each relevant track. When in due course we extend our model to predict tags for new tracks, synonymy also becomes a serious problem in evaluation: is a prediction of 'electro' correct for a track tagged only 'electronic'? And what about a prediction of 'sad' for a track actually tagged 'depressing'? While the issue of polysemy, for example 'progressive [jazz]' vs 'progressive [rock]', arises mainly from the decision to index individual words rather than entire tags, noise is inevitable due to the high subjectivity of many tags and the inherent weak semantics of music. In occasional cases, noise can also be created by explicit spam tagging of artists who are unpopular with a significant section of a particular tagging community.

Latent semantic models can mitigate all these issues by learning from co-occurrences of words over the entire collection: intuitively we learn that 'electro' co-occurs frequently with 'electronica', 'sad' with 'depressing', 'progressive' with 'rock' and 'floyd' or with 'jazz' and 'miles' (but not both at once), etc. The simplest and best-known model of this kind is so-called Latent Semantic Analysis (LSA) [Deerwester *et al.*, 1990].

In LSA, term vectors for tracks are mapped to a lower-dimensional space based on a Singular Value Decomposition of the track-term matrix for a given collection of tracks. We compute a rank-k approximation of $\mathbf{N}$

$$\tilde{\mathbf{N}}_{\mathbf{k}} = \mathbf{U}_{\mathbf{k}}\mathbf{\Sigma}_{\mathbf{k}}{\mathbf{V}_{\mathbf{k}}}^T \tag{2.5}$$

where $\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ with $\mathbf{U}^T\mathbf{U} = \mathbf{V}^T\mathbf{V} = \mathbf{I}$, $\mathbf{\Sigma}_{\mathbf{k}}$ contains the first k singular values of $\mathbf{N}$, and $\mathbf{U}_{\mathbf{k}}$ and $\mathbf{V}_{\mathbf{k}}$ the corresponding eigenvectors, for some empirically determined dimensionality k. We then base our similarities on the cosine distance between the reduced term vectors $\mathbf{U}_{\mathbf{k}}\mathbf{\Sigma}_{\mathbf{k}}$. Term vectors for queries or tracks from outside the collection are *folded in* to the latent semantic space by a simple matrix multiplication [Manning *et al.*, 2008, section 18.4]:

$$\hat{\mathbf{q}} = {\mathbf{\Sigma}_{\mathbf{k}}}^{-1}{\mathbf{U}_{\mathbf{k}}}^T\mathbf{q} \tag{2.6}$$

Retrieval with LSA is done with Algorithm 2.2, where $\hat{\mathbf{t}}$ is the k-dimensional representation of track t given by its corresponding row in $\mathbf{U}_{\mathbf{k}}\mathbf{\Sigma}_{\mathbf{k}}$. As in the simple vector space retrieval of Algorithm 2.1, the Length of track t is given by $|\hat{\mathbf{t}}|$, and the Length of the query can again be neglected.

Algorithm 2.2: LSA retrieval algorithm

```
Input: Query $q$, number $r$ of desired hits
Output: Top $r$ best matching tracks
Scores[$N$] $\longleftarrow$ 0;
Compute Lengths[$N$];
Compute $\hat{\mathbf{q}}$ by Folding In;
foreach Track $t$ do
    Scores[$t$] $\longleftarrow$ $\hat{\mathbf{q}}$ . $\hat{\mathbf{t}}$ / Length[$\hat{\mathbf{t}}$];
end
Return tracks with the highest $r$ Scores;
```

Besides solving some problems of the Vector Space Model, a low-dimensional representation for tracks and queries has the additional significant benefit of reducing the memory requirement for real world search and recommendation systems.

2.6 Aspect model

Although LSA has been applied successfully in many contexts, it has some shortcomings. In particular because it depends on a purely geometrical approach to dimension reduction, it is difficult to give any interpretation to the latent concepts that are being learned, it is uncertain in general whether they will generalise well to unseen tracks, and it is difficult to incorporate other information into the model in a principled way. Alternative probabilistic methods of dimension reduction have therefore been proposed, such as the *aspect model* or Probabilistic Latent Semantic Analysis (PLSA) introduced in [Hofmann, 1999a].

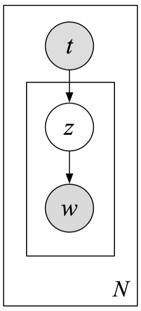


Figure 2.3: Aspect model

In the aspect model represented graphically in Fig. 2.3, we associate a latent class variable $z \in \mathcal{Z} = \{z_1, ..., z_K\}$ with each occurrence of a word $w \in \mathcal{W} = \{w_1, ..., w_M\}$ in the tags for track $t \in \mathcal{T} = \{t_1, ..., t_N\}$. The model can then be defined generatively as follows:

- select a track t with probability $P(t)$,
- select a latent class z with probability $P(z|t)$,
- select a word w with probability $P(w|z)$.

The joint probability model for the observed data is given by

$$P(t, w) = P(t)P(w|t) = P(t) \sum_{z \in \mathcal{Z}} P(w|z)P(z|t) \tag{2.7}$$

To fit the model to a collection of training tracks we maximise the log-likelihood

$$\mathcal{L} = \sum_{t \in \mathcal{T}} \sum_{w \in \mathcal{W}} n(t, w) \log P(t, w) \tag{2.8}$$

$$= \sum_{t \in \mathcal{T}} n(t) \left[P(t) + \sum_{w \in \mathcal{W}} \frac{n(t, w)}{n(t)} \log \sum_{z \in \mathcal{Z}} P(w|z)P(z|t) \right] \tag{2.9}$$

where $n(t)$ is the total number of words in tags for track t, using the Expectation Maximization (E-M) algorithm, alternating the following steps [Hofmann, 2001]:

E-step:

$$P(z|t,w) = \frac{P(w|z)P(z|t)}{\sum_{z'} P(w|z')P(z'|t)} \tag{2.10}$$

M-step:

$$P(w|z) = \frac{\sum_t n(t,w)P(z|t,w)}{\sum_{t,w'} n(t,w')P(z|t,w')} \tag{2.11}$$

$$P(z|t) = \frac{\sum_w n(t,w)P(z|t,w)}{n(t)} \tag{2.12}$$

We avoid overfitting the training data by early stopping, based on the likelihood of a validation set of tracks which we hold out from the training set. After each iteration we fold in the validation tracks to learn their aspect probabilities $P(z|t)$. Folding in is achieved as follows: we perform a fixed number of E-M iterations on $P(z|t)$ for tracks t in the validation set, following (2.10) and (2.12), but with the word probabilities $P(w|z)$ held fixed to the values learned from the main training set. We then compute the log-likelihood of the validation set according to the model, stopping when it fails to increase from the previous iteration of the main E-M process.

In practice the E- and M-steps can be interleaved, giving training a computational complexity of $O(RK)$, where R is the number of observed track-term pairs, i.e. the number of non-zero entries of $\mathbf{N}$. Algorithms 2.3 and 2.4 show this interleaved training and folding-in respectively.

Algorithm 2.3: Training an aspect model with interleaved E-M steps and early stopping

```
Input: Number of aspects $K$, vocabulary size $M$, training set of $N$ tracks,
       validation set of tracks, early-stopping threshold $\tau$
Output: Probabilities $P(w|z), P(z|t)$
Initialise $P(w|z), P(z|t)$ to random values;
Initialise accumulators $W[M][K] \longleftarrow 0$, $Z[K][N] \longleftarrow 0$;
Compute $L$ by folding in validation set;
while increase in $L > \tau$ do
    $W[M][K] \longleftarrow 0$, $Z[K][N] \longleftarrow 0$;
    foreach Track t in training set do
        foreach Word $w$ do
            foreach Aspect $z$ do
                $p[z] \longleftarrow P(w|z) * P(z|t)$;
            end
            Normalise $p[z]$ to unit sum;
            foreach Aspect $z$ do
                $W[w][z] \longleftarrow W[w][z] + n(t,w) * p[z]$;
                $Z[z][t] \longleftarrow Z[z][t] + n(t,w) * p[z]$;
            end
        end
    end
    foreach Track t do
        foreach Aspect $z$ do
            $P(z|t) \longleftarrow Z[z][t]$;
        end
        Normalise $P(z|t)$ to unit sum over $z$;
    end
    foreach Aspect $z$ do
        foreach Word $w$ do
            $P(w|z) \longleftarrow W[w][z]$;
        end
        Normalise $P(w|z)$ to unit sum over $w$;
    end
    Compute $L$ by folding in validation set;
end
```

Algorithm 2.4: Folding-in tracks into an aspect model

```
Input: Probabilities $P(w|z)$, set of $V$ tracks, number of iterations $I$
Output: Probabilities $P(z|t)$, loglikelihood $L$
Initialise $P(z|t)$ to random values;
Initialise accumulators $Z[K][V]$;
foreach iter in 1...I do
    $Z[K][V] \longleftarrow 0$;
    $L \longleftarrow 0$;
    foreach Track t in supplied set do
        foreach Word $w$ do
            $p_{tot} \longleftarrow 0$;
            foreach Aspect $z$ do
                $p[z] \longleftarrow P(w|z) * P(z|t)$;
                $p_{tot} \longleftarrow p_{tot} + p[z]$;
            end
            foreach Aspect $z$ do
                $Z[z][t] \longleftarrow Z[z][t] + n(t,w) * p[z]/p_{tot}$;
            end
            $L \longleftarrow L + n(t,w) * \log(p_{tot})$;
        end
    end
    foreach Track t do
        foreach Aspect $z$ do
            $P(z|t) \longleftarrow Z[z][t]$;
        end
        Normalise $P(z|t)$ to unit sum over $z$;
    end
end
```

To do retrieval with a trained aspect model, we first fold in a text query or track outside the training set q, following the same procedure used on the validation set, to compute its aspect probabilities $P(z|q)$. We can then use cosine distance as our matching function between the K-dimensional vectors of aspect probabilities: the retrieval algorithm is essentially identically to the one given for conventional LSA in Algorithm 2.2, but with vectors of aspect probabilities $P(z|q), P(z|t)$ taking the place of the vectors $\hat{\mathbf{q}}, \hat{\mathbf{t}}$. The formulation of the aspect model also makes it possible to use an alternative probabilistic similarity measure, estimating $P(q|t)$ directly for each track t in the collection.

2.7 Evaluating the models

We evaluate these models within a query by example framework, in particular to learn to what extent the representation of tracks within each model respects traditional catalogue organisation by artist and genre. We naturally partition our full dataset into training and test sets of tracks: to allow comparison with previous work we select the test set to replicate the experimental set-up used in a series of influential papers following [Knees *et al.*, 2004]. In these experiments artist-artist similarities were calculated for a set of 224 well-known artists split equally over 14 mainstream genres. The genre labels for each artist in this list were chosen by comparing editorial labels from the All Music Guide, Yahoo! LAUNCHcast and other sources, and can therefore be considered authoritative in comparison with individual tags [Knees, 2004]. Our corresponding test set $\mathbf{T}$ contains 1561 tagged tracks by 212 of the original 224 artists, with between 4 and 12 tracks for each artist, and 67 and 141 tracks for each genre.

In order to study the ability of our models to generalise to unseen tracks, we ensure that the training set has no artists in common with the test set. In a practical application this scenario would arise if it was undesirable to retrain the model even following the arrival of tracks by hundreds of new artists, perhaps because of computational expense or the difficulty of making updates to data used in a live search engine. More importantly, it provides a good test of whether our learned dimensions capture significant basic musical concepts,

Table 2.4: Test and training sets

Summary statistics of test set **T**, which is selected for comparison with previous work, and training set **AD**, which has no artists in common with **T**. **ADW** is a subset of **AD** containing all tracks with at least 30 words.

	tracks	**vocabulary size**	**data density %**	**% of test vocab.**
T (test)	1561	11332	0.50	100
AD (artist-disjoint)	23196	28959	0.08	67
ADW (" well-tagged)	5064	25591	0.33	62

rather than depending on artist names (which are commonly applied as social tags) and associated highly specific vocabulary. Our resulting artist-disjoint training set **AD** contains 23,196 tracks. In order to assess the effect of data sparsity (i.e. having few tags for some artists) on our models, we also select a well-tagged subset of the training set for comparison. This subset **ADW** is created by excluding all training tracks tagged with less than 30 distinct words: it contains 5,064 well-tagged tracks.

Table 2.4 shows the vocabulary sizes and data densities (i.e. the percentage of non-zero entries in the corresponding track-term matrix) for the test and training sets, after tokenizing tags for all tracks with a standard stop-list . The Table also shows the proportion of the vocabulary applied to tracks in the test set which occurs in each training set. We observe that a third of the words applied to the test tracks do not occur at all in tags applied to the training tracks: this shows the extent of the artist-specific vocabulary which we exclude when learning models from the training sets. This makes us reasonably confident that models learned on the training set which continue to perform well on the test set have indeed captured some genuine underlying semantics of tags for music.

As a baseline we use a simple vector space model with tf-idf weights on the full co-occurrence matrix for the test set, as described in Section 2.4. We then apply LSA at a range of ranks to the test and training sets, folding in the test tracks as required to create a series of models as described in Section 2.5. Finally we train a series of aspect models on the test and well-tagged training sets, again folding in the test tracks, as described in Section 2.6. For all the models we then evaluate a query-by-example search over the test set, using each test track in turn as a query and measuring various precision and recall statistics as described in the following section.

2.8 Results

2.8.1 Evaluation metrics

The accepted basic evaluation measures for information retrieval are *precision* and *recall* [Manning *et al.*, 2008, section 8.3]. Suppose we have found some number of tracks matching a query according to our system. We suppose that we know in advance whether each track in our collection is or is not relevant to the query. The precision is then the fraction of the retrieved tracks that are indeed relevant

$$precision = \frac{\#(relevant\ tracks\ retrieved)}{\#(retrieved\ tracks)} \tag{2.13}$$

and the recall is the fraction of the relevant tracks in the collection that have been retrieved

$$recall = \frac{\#(relevant\ tracks\ retrieved)}{\#(relevant\ tracks)}. \tag{2.14}$$

The precision and recall clearly will vary from query to query, and will also depend on the number of retrieved tracks we consider. In a typical web search application scenario the user will only be interested in the first few results, even though the search system may be able to find large numbers of matching items. It is therefore usual to report the precision and recall *at rank* r, often written as *precision* @r, where r is some suitable small number, meaning the precision and recall of a system averaged over a set of test queries, considering only the top r items retrieved for each query [Manning *et al.*, 2008, section 8.4].

Retrieval in the domain of music can be different in character from general

Table 2.5: Evaluation metrics

precision @5 $= \frac{2}{5}$	$\begin{array}{c}1\\0\\1\\0\\0\\\hline 1\\0\\1\\0\\0\\0\end{array}$	r-precision $= \frac{2}{4} = \frac{1}{2}$	$\begin{array}{c}1\\0\\1\\0\\\hline 0\\1\\0\\1\\0\\0\\0\end{array}$	mean AP $= \frac{\frac{1}{1}+\frac{2}{3}+\frac{3}{6}+\frac{4}{8}}{4}$ $= \frac{2}{3}$	$\begin{array}{c}1\\\hline 0\\1\\\hline 0\\0\\1\\\hline 0\\1\\\hline 0\\0\\0\end{array}$

web search. In particular we will frequently be interested in more than the top few results, for example when choosing tracks for radio streaming or playlist generation. In general the results below therefore show the per-word *mean Average Precision* (mAP), averaged over the sets of artist and genre labels. The AP for a particular query is calculated as

$$\text{AP} = \frac{\sum_{r=1}^{N} P(r)\text{rel}(r)}{\sum_{r=1}^{N} rel(r)} \tag{2.15}$$

where $P(r)$ is the precision at rank r, $\text{rel}(r)$ is 1 if the track at rank r is relevant (i.e. is labelled with same genre/artist as the query) and 0 otherwise, and N is the total number of tracks in the collection [Manning *et al.*, 2008, section 8.4]. AP therefore measures the precision averaged over the ranks at which each relevant track is retrieved. The mAP for a particular genre or artist label is the AP averaged over all queries labelled with that term, and the overall per-word mAP is the mean mAP over all the terms in the query vocabulary. Besides being a standard IR performance metric (which has become consensual in parallel literature in the field of image retrieval), mAP rewards the retrieval of relevant tracks ahead of irrelevant ones, and is consequently an extremely good indicator of how the semantic space is organised by each model.

We also report two other evaluation metrics for comparison with previous work. The *r-precision* is the precision at rank r, where r is the number of rele-

Table 2.6: Vector space retrieval compared to classification baselines

model	**genre**	**212 artists**	**32 artists**
tag vector space	0.93	0.64	0.92
web-mined text [Knees *et al.*, 2004]	0.87		
audio content-based [Mandel *et al.*, 2006]			0.68

vant tracks in the collection; note that at this rank the recall and precision are equivalent. The *Leave One Out 1-nearest neighbour classification rate* is reported in some experiments where the queries are tracks in the collection. It is calculated by inspecting the top search result returned, other than the query itself: the classification rate for a set of queries with a particular artist or genre label is the proportion of times this result also matches the label. Assuming that the query track itself is returned as the top search result, this is equivalent (within a constant) to the precision at rank 2.

Table 2.5 illustrates how the various evaluation metrics are calculated once all the tracks in a collection have been ranked by a retrieval algorithm for a particular query. Alternate columns of Table 2.5 show the pattern of relevant and irrelevant tracks in a toy collection of tracks that has been ranked by an algorithm, with the first result returned by the algorithm at the top: 1 indicates a genuinely relevant track while 0 indicates one that is not relevant. Note in particular how (despite their names) mAP and r-precision both implicitly summarise recall as well as precision.

2.8.2 Vector space model

Using our full-rank term vectors with tf-idf weighting and a cosine distance similarity measure, the genre mAP is 76%, and the artist mAP 56%. For historical reasons, many previous studies of musical similarity report a Leave One Out 1-nearest neighbour classification rate, effectively showing precision at rank 2. Table 2.6 summarises the performance of our vector space compared to these baselines. In particular the composition of our test set allows direct comparison with [Knees *et al.*, 2004], which reports a genre classification rate of 87% for a classifier trained on web-mined text. Using our vector space model to rank nearest neighbours to each query track, the equivalent LOO genre classification rate is 93%, and the *artist-filtered* classification rate, using the nearest neighbouring track by a different artist to the query, is 88%.

The LOO 1-nearest neighbour artist classification rate is 64% for our set of 212 artists. We note that this level of artist retrieval vastly exceeds the state of the art for audio content-based methods: we reach 92% precision on a reduced set of 32 rock and pop artists, compared to 68% by content-based similarity on a set of 18 similar artists in [Mandel *et al.*, 2006], although there is no reason to demand anything approaching perfect precision on this task on datasets of any size, because songs by other artists can quite reasonably be considered very similar to any given query.

2.8.3 LSA models

We show mAP retrieval performance using the LSA models over a range of ranks in Figures 2.4 and 2.5. Results are shown for models computed from the three different training sets, with the full-rank baseline shown as the right-most point in the curve for the test set $\mathbf{T}$. When LSA is applied directly to $\mathbf{T}$ it can learn the target label classes directly, as shown by the peaks in the mAP curves which coincide with the number of genre and artist labels respectively, outperforming standard vector space retrieval with the full-rank term vectors. When LSA is applied to the artist-disjoint training sets, however, retrieval performance peaks at around rank 100, suggesting that this is the underlying dimensionality of the semantic space, but retrieval performance drops significantly. The similar mAP results for $\mathbf{AD}$ and $\mathbf{ADW}$ show that there is no disadvantage in leaving sparsely tagged tracks in the training set. More importantly, however, the results show that the learned space generalises poorly to tracks by unseen artists.

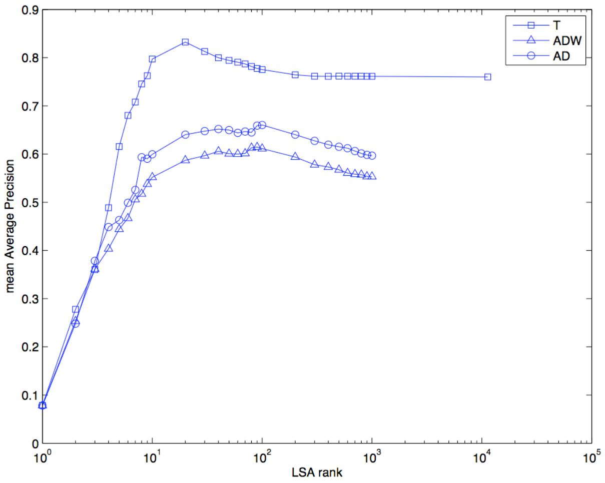


Figure 2.4: LSA genre mean AP

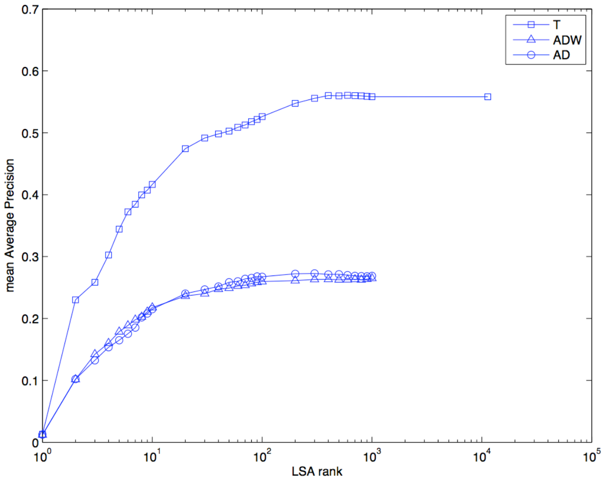


Figure 2.5: LSA artist mean AP

Table 2.7: Best retrieval results with all models, latent semantic models trained on **ADW**

model	**genre mAP**	**artist mAP**
vector space	0.76	0.56
LSA	0.61	0.25
aspect	0.75	0.49

2.8.4 Aspect models

To limit computation time when training aspect models we reduce the vocabulary size further by filtering out words applied to less than five training tracks. This reduces the vocabulary size of the **ADW** training set from 25,591 to 11,020 words, although the data density remains almost unchanged at 0.35%. The E-M training on **ADW** converged after 20-50 iterations.

Retrieval performance for the aspect models with a range of numbers of aspects is shown in Figures 2.6 and 2.7. The aspect models trained on **ADW** significantly outperform LSA at all ranks. The results for artist labels show no significant difference between aspect models trained on the test set itself and those trained on the artist-disjoint training set, while for genre labels the performance is significantly improved at nearly all ranks by training on artist-disjoint tracks. We can conclude that the aspect models generalise very well to tracks by unseen artists. We observe that aspect models appear to learn well despite the low data density of the training set, which is well below the values reported as necessary for effective training of models for collaborative filtering data reported in [Popescul *et al.*, 2001].

Table 2.7 summarises the best retrieval performance from the results shown in Figures 2.4-2.7 for both types of latent semantic model trained on **ADW** and the baseline model, showing that the simple vector space model still outperforms latent semantic models on this task. We would expect, however, that semantic models would have advantages for more realistic retrieval tasks such as query-by-example and keyword retrieval: this hypothesis is tested (and strongly confirmed) in due course in Chapter 5.

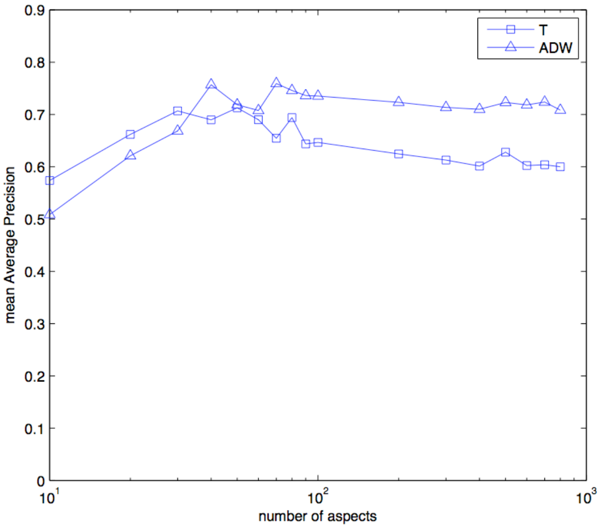


Figure 2.6: Aspect model genre mean AP

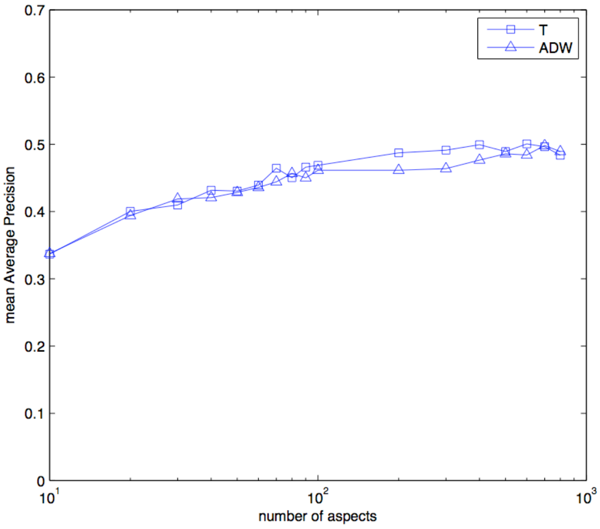


Figure 2.7: Aspect model artist mean AP

2.9 Emergent semantics

Given a trained aspect model, we can inspect the semantics of a latent aspect z directly by looking at the top-ranking words w when ordered by $P(w|z)$. The top few words for each aspect of the 90-aspect model learned from the **ADW** training set, together with their conditional probabilities, are shown in Tables 2.8-2.10. We have chosen the number, and occasionally the order, of top-ranking words shown for each aspect to give a brief meaningful description of its semantics, although to avoid over-interpretation we always show all words with an aspect-conditional probability greater than 0.1.

We observe that the aspects can be grouped fairly easily by their semantics, as shown by the headings in Tables 2.8-2.10. Genre is highly dominant, apparently accounting for over 60 of the 90 aspects. As one might expect, general *rock* and *pop* account for several aspects each, mainly relating to particular subgenres and artists, but also, in the case of pop music, corresponding to *romantic ballad*s as opposed to *fun upbeat male vocalists*. Mood itself accounts unambiguously for only one *sad* aspect. This is suprising since emotion words occur very frequently in social tags for music (as shown below in Chapter 6), suggesting that mood words may be strongly correlated with genre words. Other aspects express obviously useful concepts such as era and nationality, a few appear to be dominated by words associated with particular artists, while three aspects capture general notions of mild preference such as *favourite* and *good*, which are also very common in tags. Only three aspects, headed Tag-specific in Table 2.10, have clearly arbitrary semantics, simply capturing the co-occurrence of words found in idiosyncratic multi-word tags that happen to be frequently applied in our dataset: 'i am a party girl here is my soundtrack', 'my secret spy' and 'malloy2000 playlist - top songs - classical to metal'.

The effectiveness of these models in organising tracks in accordance with external editorial genre labels suggests that genre is well characterized by the word distributions of the learned aspects. The small number of aspects representing other music-specific concepts, however, may limit the usefulness of tag-based semantic models as the basis for music discovery and recommendation systems. The Pandora music recommendation service [6], widely regarded as being the best of its kind, is built on expert annotations using a vocabulary which is rich in such concepts, describing instrumentation, rhythmic character, harmonic complexity, etc. For comparison we also inspected a larger model with 500 aspects. Although this had learned a richer set of aspects centering on mood (expressing melancholy, dark and silent intensity, happiness, relaxation, humour, aggression, fun, high energy, dreaminess, romance, feeling good) and context (music for getting drunk to, for rainy days and coffee breaks), there was no increase in the number of aspects centred on musical concepts besides genre.

2.10 Conclusions

Although the usage of individual tags is ad hoc and informal, frequently expressing free personal responses to music rather than any attempt at collaborative structured description, using latent semantic models we can uncover an emergent semantics from social tags for music. This semantics currently focuses largely on genre, and defines an underlying similarity space for tracks that is highly organised by both genre and artist. Traditional LSA cannot learn this space effectively, overfitting the particular artists found in the training set, but, despite low data density (0.35% on our dataset), the semantics can be learned by a simple probabilistic aspect model. In subsequent Chapters the aspect model is extended to incorporate information from audio content, and its perfomance is evaluated on a wider range of retrieval tasks.

[6] http://www.pandora.com

Table 2.8: Learned semantic aspects

Genre
alternative (0.903637)
big (0.153041) beat (0.159211)
blues (0.472954) rock (0.243667)
chillout (0.224984) electronica (0.136519) ambient (0.135075) downtempo (0.0708973)
country (0.103101) love (0.167323)
electronic (0.141172) acid (0.0405934)
electronic (0.580491) electronica (0.183323)
electronic (0.137179) idm (0.136875)
experimental (0.307966) avant (0.12498) garde (0.121019)
female (0.491881) vocalists (0.346216)
female (0.234413) vocalists (0.174914) singer (0.0793115) songwriter (0.0762832)
folk (0.536908)
hardcore (0.275105) punk (0.266487)
hip (0.339388) hop (0.387366)
hip (0.327094) hop (0.336327)
hip (0.195971) hop (0.198069) rap (0.175463)
indie (0.55891)
indie (0.62218)
industrial (0.361554)
instrumental (0.365454) new (0.112528) age (0.0926655)
jazz (0.164102) fusion (0.133837)
jazz (0.453036) acid (0.092242)
vocal (0.18561) jazz (0.14978) easy (0.0727063) listening (0.0701247)
latin (0.219131)spanish (0.0853835) world (0.0823915) easy (0.0911382) listening (0.0807576)
metal (0.592658) nu (0.144499)
metal (0.443665) rock (0.144781) heavy (0.0922799)
motown (0.116136) old (0.0929515) school (0.102976) oldies (0.0968027) 60s (0.0962359)
new (0.286276) wave (0.228912) 80s (0.198137)
pop (0.395282) favorites (0.16737) favorite (0.0990058)
pop (0.699179) love (0.0434928) romantic (0.0208581) ballad (0.01479)
pop (0.534512) male (0.0902425) vocalists (0.0438629) fun (0.0477448) upbeat (0.0261533)
pop (0.213411) soft (0.160436) rock (0.1181)
post (0.425698) experimental (0.138828)
progressive (0.4349) rock (0.355539)
psychedelic (0.340054) rock (0.351364)
psychedelic (0.31107) progressive (0.112553) rock (0.110226)
reggae (0.308941) ska (0.264064)
rnb (0.192611) dance (0.124039)
classic (0.425913) rock (0.365819)
guitar (0.375608) rock (0.307403)
punk (0.661967) rock (0.102884)
rock (0.396519) alternative (0.174497)
rock (0.486211) alternative (0.0923912) american (0.08528)
rock (0.357265) alternative (0.19367) 90s (0.18555)
rock (0.253884) alternative (0.145053) political (0.0661571)
rock (0.239378) classic (0.151553) male (0.0754704) vocalist (0.0787315)

Table 2.9: Learned semantic aspects (cont.)

Genre (cont.)
rock (0.43024) deutschrock (0.0637865)
rock (0.456883) gothic (0.10763) glam (0.103065)
rock (0.229483) hard (0.126003) alternative (0.109098)
rock (0.712857) hard (0.0910012) classic (0.0853406)
rock (0.483282) indie (0.257368) alternative (0.118185)
rock (0.339735) indie (0.163039) 00s (0.169082) alternative (0.148359)
rock (0.311591) top (0.077397) song (0.0742493) radio (0.0729789)
rock (0.13689) n (0.111813) roll (0.134409) 70s (0.110408)
rap (0.391534) hip (0.0966936) hop (0.0798972)
singer (0.380835) songwriter (0.372339)
singer (0.294251) songwriter (0.285342) folk (0.166677)
soul (0.3935) rnb (0.0829992)
synth (0.113573) pop (0.115882) synthpop (0.106553)
trance (0.151609) australian (0.0702906) chilled (0.0693821)
trip (0.29392) hop (0.269011)
Nationality
british (0.843287)
britpop (0.17749) indie (0.209654)
french (0.133541) dance (0.243374)
german (0.265269) deutsch (0.127166)
irish (0.208149) rock (0.273498)
swedish (0.104755) alternative (0.161654)
uk (0.223606) english (0.122075) england (0.107208)
Era
60s (0.241197) rock (0.310489) classic (0.199013) oldies (0.101261)
70s (0.332746) male (0.0538104) faves (0.0407649) great (0.0291682) rolling (0.0275969) stones (0.0290014)
80s (0.365832) rock (0.0922272)
90s (0.829352)
00s (0.21898) drum (0.141555) n (0.0411199) bass (0.199598)
Other musical
cover (0.232786) covers (0.155134) ballad (0.113401)
piano (0.277365)
soundtrack (0.446551)
Artist
alternative (0.189528) female (0.140492) icelandic (0.0778212) vocalists (0.0733118) bjork (0.0258379)
beatles (0.19894) pop (0.107256) rock (0.10571) british (0.0833975) invasion (0.0518532) lennon (0.0246465)
funk (0.399733) red (0.0751185) hot (0.0812188) chili (0.067166) peppers (0.0689253)
rock (0.822775) muse (0.01499) pink (0.0146779) floyd (0.0135611)
rock (0.141085) songs (0.0562564) queen (0.0399782) classic (0.0306823)

Table 2.10: Learned semantic aspects (cont.)

Mood acoustic (0.342529) mellow (0.170217) chill (0.0807014) sad (0.1822) melancholic (0.125459) beautiful (0.100356) melancholy (0.0854095) mellow (0.0839701)
User-specific seen (0.237975) live (0.327781) world (0.118652) music (0.033688)
Preference favourite (0.279855) songs (0.135857) best (0.109118) artists (0.1018) ever (0.091739) favorite (0.0802845) favourite (0.0949422) songs (0.114899) essential (0.0912613) cool (0.0885247) good (0.134277) love (0.116914) male (0.111679) vocalist (0.0653571) favorites (0.0807613)
Tag-specific i (0.111074) am (0.105064) party (0.126584) girl (0.104435) my (0.116596) soundtrack (0.107144) my (0.181792) secret (0.107245) spy (0.109136) top (0.297385) songs (0.156311) malloy2000 (0.131718) playlist (0.139019) classical (0.15024)

Chapter 3

A discrete representation for musical audio features

This Chapter proposes a discrete representation for musical audio features that allows the models of the preceding Chapter to be extended in a straightforward fashion to audio content as well as words in tags. Much work in recent years has focussed on developing low-level features intended to capture musically meaningful aspects of an audio signal, in particular in the hope of doing reliable automatic genre classification. Discrete representations of audio of the kind that would be useful to us here, however, have been used only as an approximate and poorly-performing computational shortcut. The remainder of this Chapter discusses this work in detail, and motivates, describes and evaluates a new representation that can easily be used in a conventional information retrieval framework. It begins, however, by discussing why we should be interested in modelling audio content at all, given the increasing availability of tags.

Why model audio content?

Chapter 2 demonstrated that semantic models learned from social tags have highly attractive properties. Even the simplest vector space models position tracks in a space which is extremely well-organised by artist and genre, while latent semantic models can learn a wide range of familiar and readily meaningful semantic aspects. It is reasonable to speculate that, as long as tags are readily available for all the tracks we wish to index, audio information is redundant if we want to create practical semantic music search applications. Tags are being supplied by listeners in huge numbers: last.fm currently receives around two million new annotations each month.[1] There are reasons, however, why despite the huge and growing number of tags available, the distribution of social tags is likely to remain highly uneven in practice, meaning that we should expect to find many sparsely-tagged or untagged tracks in any large collection.

Firstly, new music is constantly being created, leading to the well-known *cold start* problem: tracks can be tagged only once listeners discover them, but untagged new tracks remain invisible within systems that depend on tags to give search results or recommendations. Secondly, recent research [Marlin *et al.*, 2007] has highlighted the correlation between a listener's liking for a particular track and their willingness to supply a rating for it: listeners are much more likely to rate a track which they like or (somewhat less often) dislike strongly. Ratings for tracks that are new to a particular listener are therefore *not missing at random* (NMAR), contradicting an underlying assumption of most existing collaborative filtering systems. We can expect a similar relationship to exist for tagging, with tracks that provoke only mild feelings of affection in their listeners remaining sparsely-tagged, even if they have obvious characteristics that could be described in words. In particular we expect that there will be a clear difference between the distribution of tags for tracks by mainstream and by new or niche artists.

This uneven distribution of tags between 'haves' and 'have-nots' can be

[1] private communication from Elias Pampalk, last.fm, March 2008

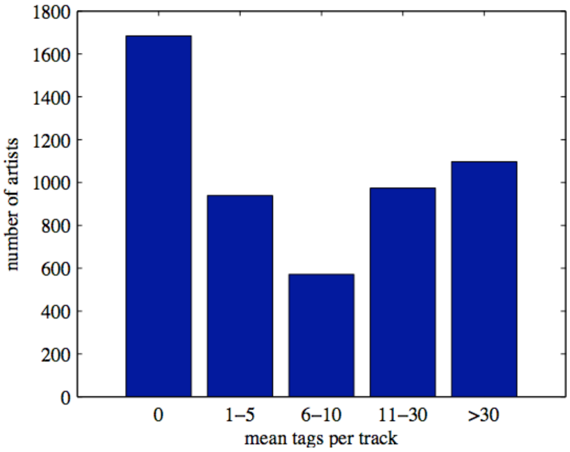


Figure 3.1: Artist tag distribution

clearly observed in our dataset, as illustrated in Figure 3.1, which shows the number of artists found in our dataset as a function of the mean number of tags applied to their tracks. Roughly a third of our 5265 artists have received no tags for any of their tracks, while even amongst the artists with tagged tracks, roughly a third have no more than five distinct tags per track on average. The cold start and NMAR issues evident here will give real-world music recommendation or search systems based on tags an inbuilt conservative bias towards tracks by well-known and well-liked artists. While this is a reasonable starting point for a usable system, the ability to suggest a large variety of tracks, in particular including little-known music, is clearly also valuable. This provides a practical motivation to extend our models by incorporating information drawn directly from the audio signal. It also suggests a realistic framework for evaluating the contribution of such audio information to both the quality and variety of results returned to set of search queries: we develop this in Section 4.2.

One straightforward way to incorporate audio information into semantic models is to discretise audio features, representing them as a set of “audio words” extending, or parallel to, the vocabulary of conventional words. A simple method of this kind, using vector quantisation (VQ) to discretise the features and treating the resulting VQ codebook as the vocabulary of audio words, was first proposed in a somewhat different context by Vignoli and Pauws in [Vignoli & Pauws, 2005], where a discrete representation was chosen as the basis of a similarity metric for audio tracks because of its computational efficiency in relation to existing methods. In [Vignoli & Pauws, 2005], a single Self-Organising Map (SOM, defined below in Section 3.2.1) trained on features drawn from all tracks in the collection to be indexed was used for VQ. Features from each track were mapped onto the indices of their best-matching SOM units, and the indices for each track recorded in a histogram. A distance between tracks could then be computed by comparing histograms with a suitable measure: Vignoli and Pauws proposed Kullback-Leibler divergence.

We investigated this representation in comparison to a number of other lightweight audio similarity measures in previous work published in [Levy & Sandler, 2006b]. Despite finding a more effective distance measure to compare the histograms than that used by Vignoli and Pauws, our results showed that tracks were poorly organised in the resulting similarity space: in particular using this discretisation degraded results in comparison to similarity measures computed directly on the underlying features. In this chapter we propose a new approach to extracting a discrete vocabulary of audio *muswords* intended to correspond to properties of important musical events within each track. We show in particular that tracks in our test dataset are no worse organised by muswords in a simple vector space model than when using a state-of-the-art similarity measure directly on the features.

Audio features intended to model perceptual characteristics of music have been widely studied in the context of automatic genre classification, with features for a particular track typically modelled as a so-called *bag-of-frames*, i.e. all frames in the track are modelled but with no consideration of their temporal sequence. While the bag-of-frames (BOF) model works well for classification of non-musical audio such as natural ambient soundscapes, detailed studies by Aucouturier in [Aucouturier *et al.*, 2007] and [Aucouturier, 2006] highlight its shortcomings in relation to music. In particular Aucouturier observes ([Aucouturier *et al.*, 2007], p.889) that

> with BOF algorithms, frames contribute to the simulation of the auditory sensation in proportion of their statistical predominance in the global frame distribution. In other words, the *perceptive saliency* of sound events is modeled as their *statistical typicality*... The above-presented results establish, as expected, that the mechanism of auditory saliency implicitly assumed by the BOF approach does not hold for polyphonic music signals: For instance, frames in statistical minority have a crucial importance in simulating perceptive judgments.

Aucouturier hypothesises that higher-level features are required to improve classification performance on musical audio. In our work, this problem is compounded by the obvious mismatch between semantics and either individual audio frames or track-level models. While fully addressing these issues remains well beyond the scope of this thesis, we use them to motivate a novel approach to audio feature modelling, based on an initial step in which we identify regions of interest within each track.

We make the following simple assumptions:

1. semantics apply naturally to music at the phrase level (a single track can contain both *harsh* and *gentle* sections)

2. semantics are associated with particular events within the music (rather than with individual audio frames)

3. significant musical events will be perceptually prominent by design (both composer and performer devote their skill to bringing this about)

We consequently extract muswords for a track by first identifying musical events within it, and then discretising timbral and rhythmic features for each region found.

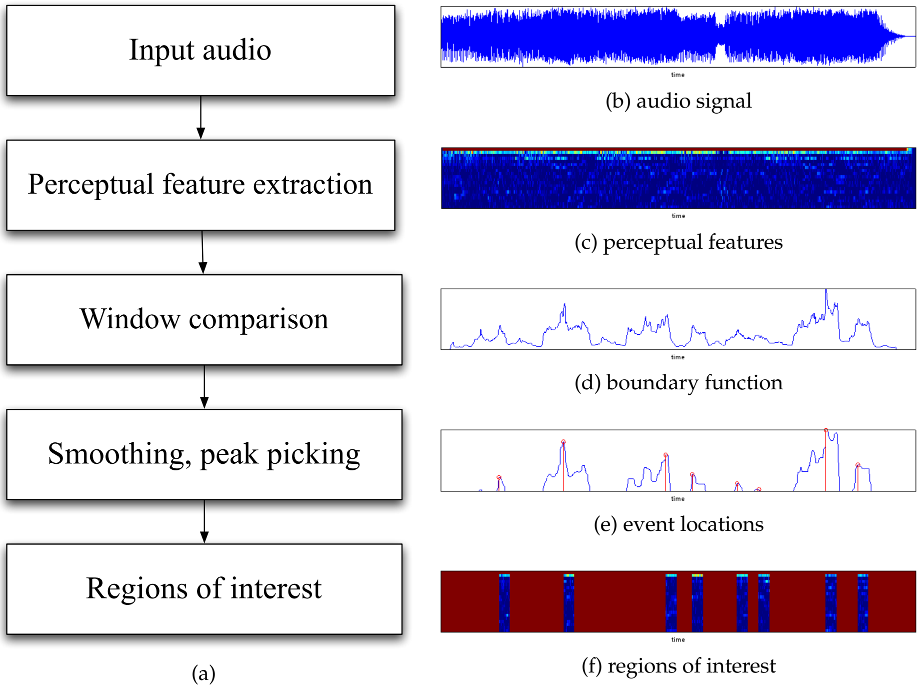
We note that this perspective differs from previous work on semantic music search and annotation, in which semantics are associated either with every frame of audio [Barrington *et al.*, 2007; Turnbull *et al.*, 2008] or with randomly selected segments [Eck *et al.*, 2008].

The remainder of this chapter is organised as follows: Section 3.1 introduces a method for finding regions of interest within each track; Section 3.2 shows how timbral and rhythmic features from each region are mapped onto discrete muswords corresponding to musical properties of the audio signal; and Section 3.3 evaluates musword representation in a simple vector space model, and conclusions are drawn in Section 3.4.

3.1 Finding regions of interest

A number of methods have been proposed to find representative *thumbnail* segments of musical audio tracks, typically based on a first step in which the repetition structure of the track is estimated [Maddage *et al.*, 2004; Lu *et al.*, 2004; Goto, 2003; Chai & Vercoe, 2003; Paulus & Klapuri, 2006; Shiu *et al.*, 2006]. We review these approaches in our own contributions to this literature [Levy *et al.*, 2006; Levy & Sandler, 2006a, 2008a] . While some of these structural segmentation algorithms have been shown to be effective in locating chorus sections in conventional pop tracks, notably [Goto, 2003], they are not suitable for our purposes here, in particular because the initial analysis of repetition structures within a track is too computationally expensive to scale to large music collections.

Assumption (3) above, on the other hand, suggests a straightforward and computationally scalable method to locate musical events by finding perceptually prominent regions of interest within the signal: such regions are identified by their degree of contrast with what has come before in the track. Figure 3.2 shows an overview of the process. We first extract perceptually-motivated audio features for the whole track. We then pass a fixed-length window along the track, comparing the distribution of features in the window to their distribution in the time-decayed history (i.e. from the beginning of the track to the start of the window) with a probabilistic distance measure. The distance of the window from its history gives us a boundary function, expressing the contrast between them, and consequently, given assumption (3), the likelihood of an event beginning at the start of the window. We smooth the boundary function with a median filter to eliminate noise from local contrast, and *peak-pick*, i.e. find local maxima in the smoothed boundary function, to give a set of candidate event start times. Finally we normalise for the degree of local contrast within each track by discarding candidates whose boundary function is less than the mean value over the whole track. We return windows beginning at each of the remaining event start times as the track's regions of semantic interest.



(a)

(b) audio signal

(c) perceptual features

(d) boundary function

(e) event locations

(f) regions of interest

Figure 3.2: Locating regions of interest
Locating regions of interest. (a): overall flowchart. (b): input audio signal. (c)-(f): outputs at each stage. (c) perceptual features (MFCCs); (d) unsmoothed boundary function; (e) smoothed, mean-subtracted boundary function and found event start times; (f) identified regions.

In our current implementation we use the first twenty Mel-Frequency Cepstral Coefficients (including the 0-th coefficient) as our perceptual audio features, extracted from audio downsampled to 22.05kHz and mixed to mono, with a frame and hop size of 4096 samples. Mel-Frequency Cepstral Coefficients (MFCCs) represent the short-term power spectrum on a non-linear frequency scale inspired by the human auditory system [Mermelstein, 1976]. They are computed as follows:

1. a Hamming window is applied to each frame of audio.
2. the Fourier Transform is taken.
3. the *mel spectrum* is computed by applying a filterbank of overlapping triangular windows centred on mel frequencies.
4. MFCCs are computed as the amplitudes of the Discrete Cosine Transform of the log powers of the mel spectrum.

The moving window used in computing the boundary function from the MFCCs has a length of 5 seconds. We estimate the distribution of MFCCs in the moving window v and the history h by fitting a single Gaussian to features in each of them, weighting features in the history with a Hamming window extending back to the start of the track, so that features from the distant past are gradually "forgotten". We measure the distance between the two Gaussians with a symmetrised Kullback-Leibler divergence

$$\begin{aligned} KL_s(v||h) &= KL(v||h) + KL(h||v) \qquad (3.1) \\ &= \frac{tr(\Sigma_h^{-1}\Sigma_v + \Sigma_v^{-1}\Sigma_h) + (\mu_v - \mu_h)^T(\Sigma_h^{-1} + \Sigma_v^{-1})(\mu_v - \mu_h)}{2} - d \end{aligned}$$

where the Gaussians are given by $v(x) = \mathcal{N}(x; \mu_v, \Sigma_v)$ and $h(x) = \mathcal{N}(x; \mu_h, \Sigma_h)$, and d is the dimensionality of the features.

This boundary function is smoothed with a median filter of length 2 seconds. Finally after peak-picking we prune candidates that are within two window lengths of each other, retaining the one with the higher boundary function value.

Following the motivating assumptions listed in the previous Section, the temporal regions found by this process are expected to correspond to the significant musical events in the track, i.e. the regions most likely to bear semantics and to be usefully modelled for retrieval or automatic annotation. Low-level features are therefore extracted for each of these regions, and mapped onto one or more muswords representing characteristic areas of the audio feature space. The set of muswords for a track is the union of the muswords associated with each of its temporal regions of interest. The following Section proposes various ways in which audio features for a given region can be represented as muswords.

3.2 A vocabulary of audio muswords

In considering how each region of interest found in a track can be mapped onto muswords, we assume that is reasonable to create muswords representing two independent vocabularies of timbral and rhythmic characteristics respectively, i.e. we attempt to describe a musical event within a track as having on the one hand some particular type of instrumentation, and on the other some particular type of tempo and beat.

3.2.1 Creating timbre muswords

Our underlying timbral feature for each region of interest is the same feature that we used when computing the boundary function for event-finding described in the previous Section, i.e. the mean and variance of the first twenty MFCCs. This Subsection describes two alternative methods of representing these features as muswords. The methods are evaluated comparatively in Section 3.3.

VQ method

We concatenate means and variances into a single 40-dimensional feature for each region of interest. Following our work in [Levy & Sandler, 2006b], we train a single Self-Organising Map on features from our collection of tracks, first normalising each feature dimension to have zero mean and unit variance. The SOM is a simple unsupervised neural network which learns a mapping of input vectors to a very low-dimenstional grid: the mapping captures non-linear relationships between the input vectors as geometrical relationships in the grid, in particular preserving the local topology of the input vectors [Kohonen, 1984]. Each grid location or *neuron* of the SOM is associated with a weight vector m_i with the same dimensionality as the input vectors. On each training step, an input vector x is chosen at random, its *best matching unit* in the SOM is found, i.e. the neuron m_c whose weight vector is closest to the input one, and the weight vectors for that unit and those in its neighbourhood are updated to move them closer to the input vector. The update rule at time k is given by

$$m_i(k+1) = m_i(k) + \alpha(k)h_{ci}(r(k))[x - m_i(k)] \tag{3.2}$$

where $m_i(k)$ is the value of the i-th weight vector, $\alpha(k)$ is the learning rate, and $h_{ci}(r(k))$ is a neighbourhood function around the best matching unit m_c, with radius $r(k)$. Both the learning rate and the neighbourhood size decrease over time.

We use a SOM with 1000 hexagonal units arranged in a rectangular 50 x 20 grid, and a Gaussian neighbourhood function, as implemented in the SOM Toolbox [Vesanto, 2000]: each unit represents one timbre musword. A single musword for each region of interest in a track is then created by finding its best matching unit in the trained SOM.

Distance method

A simple perceptual test was applied to the mapping to muswords using the VQ method described above. A sample of 50 muswords was chosen at random and considered in turn. For each musword, a sample of 20 audio segments that mapped onto it was concatenated. Finally we simply listened back to the patchwork of audio segments for each musword. The results were disappointing: for many muswords there appeared to be little perceptual timbral consistency between the regions. We therefore developed an alternative mapping based closely on the timbral distance measure in (3.1), which is known to be relatively well-behaved [Levy & Sandler, 2006b].

We first select 1000 regions of interest at random from our collection of tracks, and consider these directly as comprising our vocabulary of timbre muswords. We then map a region of interest with features x not onto integer counts, but instead onto a vector of continuous relevance scores, $\{c(x, m)\}$ with $c(x, m) \in (0, 1], \forall m$, based on the distance of the region to each musword m in the vocabulary. The score for the musword m for a region with features x, is given by

$$c(x, m) = \frac{1}{(1 + KL_s(x||y_m))} \tag{3.3}$$

where y_m are the features for musword m, and the distance measure $KL_s(\cdot||\cdot)$ is the symmetrised Kullback-Leibler divergence given in (3.1). Finally we compute the relevance scores for a track $\{c(t, m)\}$ by summing the scores for each musword over all of the track's regions of interest.

Because each region of interest is mapped onto a score for every musword in the timbre vocabulary, in general this representation is no longer sparse. This will prove a disadvantage in our aspect models, where the computational complexity is proportional to the total number of non-zero (mus)word counts over the training set, as discussed in Section 2.6 above (and in fact the same applies to industrial-scale implementations even of simple vector space models). We therefore increase sparsity by zeroing small scores for timbre muswords in this representation. Specifically we set scores for track t to zero when they are less than $\sigma \max_m c(t, m)$, where $c(t, m)$ is the track's total relevance score for musword m. We discuss the choice of the threshold σ in the next Section.

3.2.2 Creating rhythm muswords

Our rhythmic feature for each region of interest is the thresholded autocorrelation of an onset detection function introduced by Davies and Plumbley in [Davies & Plumbley, 2008]:

$$A(l) = \frac{\sum_{l'=1}^{L} \tilde{\Gamma}(l')\tilde{\Gamma}(l'-l)}{|l-L|} \quad l = 1, ..., L \tag{3.4}$$

where $L = 144$ samples and $\tilde{\Gamma}(\cdot)$ is an adaptively-thresholded onset detection function based on complex spectral difference (see [Davies & Plumbley, 2008] and [Bello *et al.*, 2004] for full details). This feature was found in [Davies & Plumbley, 2008] to give good results in a classification task for different styles of ballroom dance music.

We follow the VQ approach as for timbre muswords, training a 50 x 20 SOM on these 144-dimensional features and mapping each region of interest onto its best matching unit. Unlike the timbre muswords produced by VQ, this approach does satisfy a simple perceptual test: in informal listening tests we found that regions mapped to the same unit frequently have the same tempo and rhythmic character.

3.3 Evaluating the bag-of-muswords

The methods of the previous Section produce a bag-of-muswords (BOM) for each track. We evaluate this representation initially in a simple Vector Space model, just as we did for the tag BOW in Chapter 2. Audio was not available for all 1561 tracks in our test set **T**, so we pruned it to create a reduced set of 928 tracks with audio **Ta**, with between 25 and 98 tracks for each of the 14 labelled genres. We evaluated artist retrieval over the remaining 105 artists with at least 4 tracks each in **Ta**. The results in this Section are all based on query by example over the tracks in this set, using a Vector Space model with tf-idf weighting, with document frequencies for each musword computed over the test set. We prune from the vocabulary any muswords applied to less than five tracks in the set.

3.3.1 Sparsifying the distance method timbre muswords

Figure 3.3 shows retrieval results using timbre muswords created by the distance method of Section 3.2.1. This illustrates the effect of sparsifying the continuous relevance scores produced by this method to varying degrees, by zeroing all scores for each track which are less than some proportion σ of the score for its most relevant musword. We can clearly reduce data density to under 10% with no significant loss in retrieval performance: in practice we set $\sigma = 0.6$, which gives a data density of 7.4% on the test set.

3.3.2 Results

Table 3.1 gives average genre and artist retrieval precision figures using each track in the test set as the query in a query by example scenario. Besides the mean Average Precision (mAP) reported in Section 2.7, Table 3.1 shows the precision at rank 5 for genre labels, and the r-precision for artist identity, i.e. the precision at rank r, where r is the total number of tracks by the query artist in the collection. These two figures give a measure of the performance at high ranks, reflecting the results that would be seen in practice by the user of a search engine, while the mAP figures express the quality of organisation over the entire collection. The best BOM results are shown in bold.

Besides comparing the BOM with timbre muswords created by the VQ and distance methods described in the previous section, we give results for three baseline methods. For our primary baseline we evaluate content-based

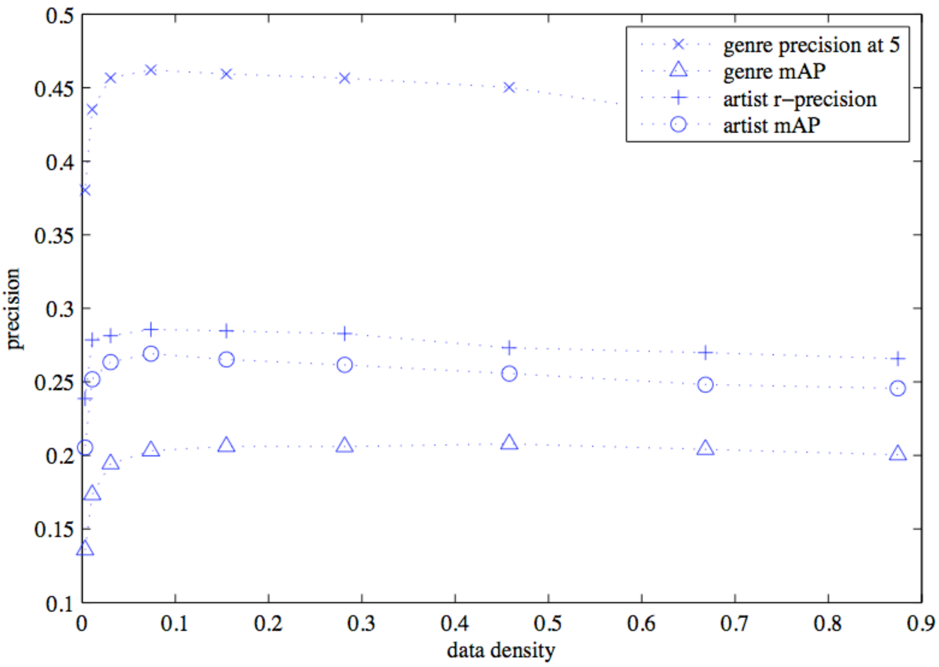


Figure 3.3: Retrieval performance vs data density
The sparsification threshold σ takes values 0.9, 0.8, ... 0.1

retrieval using a state-of-the-art distance measure directly on the underlying timbral audio features: we use symmetrised Kullback-Leibler divergence on single Gaussians fitted to MFCCs from the whole of each track [Mandel & Ellis, 2005; Levy & Sandler, 2006b]. We also show results for a random baseline, and for the BOW Vector Space model re-evaluated on the reduced test set.

The results in Table 3.1 show that timbre muswords created by the distance method of Section 3.2.1 are significantly more effective than those created by VQ. The organisation of our test tracks in a simple BOM model using these muswords is similar to using a state-of-the-art similarity measure directly on the underlying features: genre retrieval is marginally better in the BOM model, and artist retrieval slightly worse. Rhythm muswords, however, give poor retrieval performance on their own, and either make no significant difference or reduce performance when combined with timbre muswords.

Table 3.1: BOM retrieval performance

	genre prec. at 5	**genre mAP**	**artist r-prec.**	**artist mAP**
BOM:				
rhythm	0.322	0.121	0.233	0.203
VQ timbre	0.387	0.168	0.251	0.228
VQ timbre + rhythm	0.379	0.165	0.247	0.227
distance timbre	**0.462**	**0.203**	**0.286**	**0.269**
distance timbre + rhythm	0.439	0.196	0.278	0.256
baseline:				
random	0.262	0.099	0.208	0.175
timbre similarity	0.461	0.187	0.304	0.288
BOW	0.939	0.774	0.581	0.629

3.4 Conclusions

This Chapter introduced a method of finding regions of interest within a track that - while only a first simple implementation of the approach - leads to an effective discretisation of audio as a vocabulary of timbral *muswords*. Query by example using these muswords is more successful than with previous discrete representations, equalling the performance of an effective similarity measure applied directly to the underlying audio features. Rhythm muswords, while inducing some organisation on the collection when compared with a random baseline, unfortunately do not improve retrieval performance when combined with timbre muswords, and are therefore not used in the models developed in the course of the following Chapter.

The novel musword representation developed here makes it straightforward to extend retrieval models to audio content as well as words. Nonetheless, the most striking result in Table 3.1 is the difference in performance between the baseline model trained on words and any of the audio content-based methods. This raises issues in evaluating the contribution of audio features to joint models: we return to this in the next Chapter.

Chapter 4

Learning semantic models for music from social tags and audio

This Chapter extends the aspect model of Chapter 2 to incorporate muswords as well as words, providing the basis of a search system that learns from both social tags and audio. The first step towards this is to combine words and muswords in a single Vector Space model. We can take a straightforward approach here, simply concatenating words and muswords into a single extended vocabulary, so the track representation is a bag-of-words-and-muswords (BOW+M). Because of course we are not really counting words in documents, we observe that "counts" for the two types of word in this representation, $n(t, w)$ and $c(t, m)$ in the notation of the preceding Chapters, have dissimilar - and essentially arbitrary - ranges. A consequence of this is that it is necessary to choose a scaling for counts for muswords relative to those for words.

The results of Chapter 2 demonstrate that retrieval models based on tags place tracks in a space which respects traditional catalogue organisation extremely well, in fact outperforming all previous published methods on genre and artist retrieval tasks. We also know, however, that in real music collections tracks by many artists will be at best sparsely tagged. This Chapter attempts to establish the level of tag sparsity at which retrieval based purely on words starts to degrade, and investigates whether or not combining words and muswords does indeed improve performance according to objective measures. This leads to a realistic cross-validation framework for evaluating joint models trained on tags and audio features.

Finally this Chapter investigates how latent aspect models can best be trained on words and muswords, again given the much higher reliability of drawing semantic information from tags than inferring it from current low-level audio features. Two different training strategies are compared for these models: conventional training on the joint vocabulary of words and muswords, and a two-stage training method, in which we first learn the latent aspects from words only, and then learn the musword distributions.

The remainder of this Chapter is organised as follows: Section 4.1 discusses how words and muswords can be combined into a joint vocabulary by scaling counts; Section 4.2 details a framework for evaluating the contribution of muswords to semantic search; and Section 4.3 investigates the effect of tag sparsity on track organisation in a Vector Space model based on the joint vocabulary. Two training methods for a joint aspect model are explained in more detail in section 4.4; section 4.5 describes how the evaluation developed in the Section 4.2 is applied to the resulting models; results are given in section 4.6.

4.1 Scaling word counts

In the BOW+M representation, counts for words $n(t, w)$ and muswords $c(t, m)$ are computed by different means, and have no natural scaling with respect to one another. Specifically the counts for conventional words depend on Last.fm's unpublished normalisation of the number of times a tag has been applied to any particular track, as described at the start of Chapter 2, while the

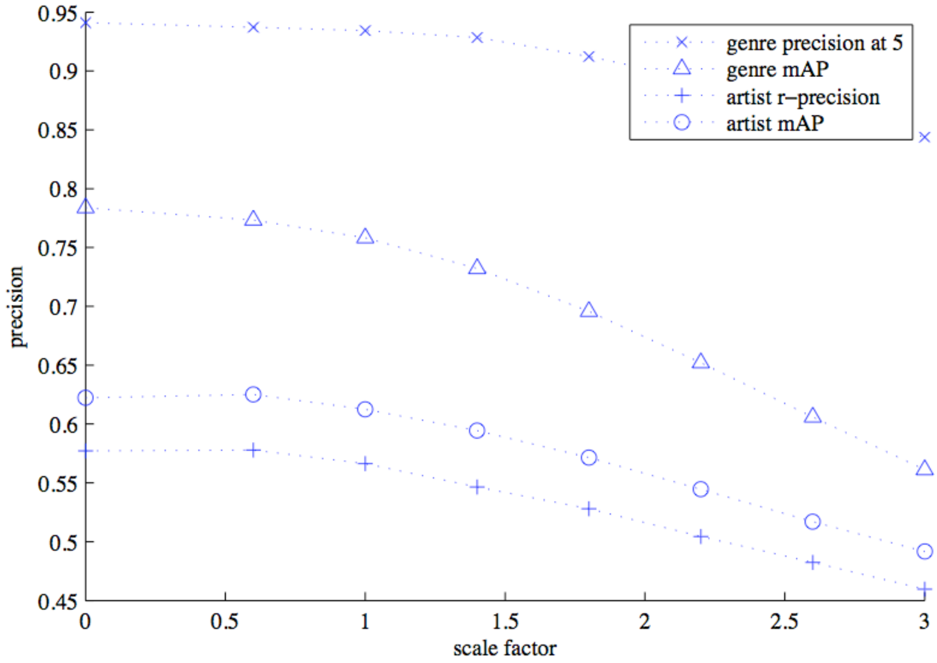


Figure 4.1: BOW+M retrieval performance

counts or continuous scores for muswords result from the particular discretisation method used to map features for a track onto muswords, specifically the methods described in Chapter 3.

Figure 4.1 illustrates how the relative scaling of word and muswords counts affects retrieval performance, using the same tasks and evaluation metrics as the experiments of Sections 2.8 and 3.3. Retrieval is done in a simple Vector Space BOW+M model with tf-idf weighting with a range of different scalings between the two sets of counts: the *scale factor* shown is the ratio between the mean count for muswords and that for words. A scale factor of zero corresponds to discarding muswords completely i.e. using a baseline BOW model. Table 4.1 gives the top ten search results returned by this model for some example query tracks at several scale factors.

Using a scale factor of 1.0, the retrieval performance is slightly lower than the BOW baseline, but, as the examples in Table 4.1 show, search results for query by example in this model are largely acceptable, although by no means

Table 4.1: Example search results

scale factor = 0.0	scale factor = 1.0	scale factor = 3.0
Joni Mitchell: Both Sides Now	**Joni Mitchell: Both Sides Now**	**Joni Mitchell: Both Sides Now**
Joni Mitchell: Free Man In Paris	Joni Mitchell: You Turn Me On I'm A Radio	Joni Mitchell: You Turn Me On I'm A Radio
Joni Mitchell: You Turn Me On I'm A Radio	Joni Mitchell: Free Man In Paris	Thelonious Monk: Thelonious
Leonard Cohen: Sisters of Mercy	Leonard Cohen: Sisters of Mercy	Herbie Hancock: Tell Me A Bedtime Story
Leonard Cohen: Story of Isaac	Leonard Cohen: Famous Blue Raincoat	Dave Brubeck: Blue Rondo a la Turk
Leonard Cohen: Famous Blue Raincoat	Pete Seeger: Little boxes	Leonard Cohen: Bird on the Wire
Pete Seeger: Little boxes	Leonard Cohen: Bird on the Wire	Steeleye Span: Gaudete
Leonard Cohen: First We Take Manhattan	Leonard Cohen: Story of Isaac	Bob Dylan: Like a Rolling Stone
Bob Dylan: Mr. Tambourine Man	Steeleye Span: Gaudete	Leonard Cohen: Everybody Knows
Steeleye Span: All Around My Hat	Bob Dylan: Blowin' in the Wind	Leonard Cohen: Famous Blue Raincoat
Radiohead: Karma Police	**Radiohead: Karma Police**	**Radiohead: Karma Police**
Weezer: No Other One	Weezer: No Other One	The Smiths: There Is a Light That Never Goes Out
Radiohead: We Suck Young Blood	Radiohead: A Wolf at the Door	Smashing Pumpkins: Disarm
Radiohead: A Wolf at the Door	Radiohead: We Suck Young Blood	Radiohead: A Wolf at the Door
Radiohead: A Wolf At The Door	Foo Fighters: Up in Arms	Weezer: No Other One
Foo Fighters: Up in Arms	Smashing Pumpkins: Disarm	The Smiths: Last Night I Dreamt...
Sonic Youth: Tunic (Song for Karen)	Radiohead: A Wolf At The Door	Radiohead: Myxomatosis
Smashing Pumpkins: Disarm	Sonic Youth: Tunic (Song for Karen)	Robbie Williams: She's the One
Weezer: Beverly Hills	Smashing Pumpkins: Bullet With Butterfly Wings	The Clash: London Calling
Jane's Addiction: Just Because	Weezer: Beverly Hills	Robbie Williams: Something Beautiful
Moby: My Weakness	**Moby: My Weakness**	**Moby: My Weakness**
Aphex Twin: Xtal	Aphex Twin: Xtal	Aphex Twin: Xtal
Moby: My Beautiful Blue Sky	Moby: My Beautiful Blue Sky	Moby: My Beautiful Blue Sky
Moby: Natural Blues	Aphex Twin: Avril 14th	Aphex Twin: Avril 14th
Moby: Sunday (The Day Before My Birthday)	Moby: Natural Blues	Aphex Twin: Kladfvgbung Micshk
Aphex Twin: Avril 14th	Moby: Sunday (The Day Before My Birthday)	Aphex Twin: Btoum-Roumada
Moby: Honey	Aphex Twin: Kladfvgbung Micshk	Underworld: Mmm Skyscraper I Love You
Aphex Twin: Kladfvgbung Micshk	Moby: Honey	Moby: Natural Blues
Kraftwerk: Computerliebe	Aphex Twin: Btoum-Roumada	Moby: Sunday (The Day Before My Birthday)
Aphex Twin: Btoum-Roumada	Underworld: Mmm Skyscraper I Love You	Wolfgang Amadeus Mozart: Agnus Dei
Sonic Youth: 'Cross the Breeze	**Sonic Youth: 'Cross the Breeze**	**Sonic Youth: 'Cross the Breeze**
Sonic Youth: Tunic (Song for Karen)	Sonic Youth: Tunic (Song for Karen)	Dead Kennedys: Holiday in Cambodia
Sonic Youth: Disappearer	Sonic Youth: Tunic	Deep Purple: The Battle Rages On
Sonic Youth: Master-Dik	Foo Fighters: Best of You	Sonic Youth: Tunic
Sonic Youth: Tunic	Sonic Youth: Disappearer	Sonic Youth: Tunic (Song for Karen)
Sonic Youth: Mary-Christ	Weezer: No Other One	Foo Fighters: Best of You
Radiohead: Karma Police	The Smiths: The Headmaster Ritual	Sepultura: Endangered Species
Weezer: No Other One	Foo Fighters: Burn Away	ABBA: So Long
Weezer: Beverly Hills	Smashing Pumpkins: Bullet With Butterfly Wings	Foo Fighters: Burn Away
Radiohead: A Wolf at the Door	Sonic Youth: Mary-Christ	The Smiths: The Headmaster Ritual
Slayer: Jesus Saves	**Slayer: Jesus Saves**	**Slayer: Jesus Saves**
Slayer: Raining Blood	Slayer: Raining Blood	Slayer: Raining Blood
Slayer: Angel of Death	Slayer: Altar of Sacrifice	Slayer: Altar of Sacrifice
Slayer: Altar of Sacrifice	Slayer: Angel of Death	Slayer: The Antichrist
Anthrax: Caught in a Mosh	Slayer: The Antichrist	Slayer: Angel of Death
Slayer: The Antichrist	Anthrax: Caught in a Mosh	Megadeth: A Tout Le Monde
Sepultura: Endangered Species	Anthrax: Got the Time	Anthrax: Madhouse
Anthrax: I Am the Law	Sepultura: Itsári	Pantera: Strength Beyond Strength
Anthrax: Got the Time	Anthrax: Madhouse	Anthrax: Got the Time
Sepultura: Itsári	Megadeth: A Tout Le Monde	Pantera: New Level

identical to those returned by searching on words only. With a scale factor of 3.0, however, objective retrieval performance is reduced significantly, and the search results include more surprises.

A more detailed examination of the examples in the third column of Table 4.1 is informative. At first glance, the jazz tracks returned for Joni Mitchell's 'Both Sides Now' are poor matches, because Mitchell is most often labelled as a folk singer (as she is in our genre groundtruth). 'Both Sides Now', however, is the title track of an album of classic jazz songs, and the pianist on the album is none other than Herbie Hancock, whose 'Tell Me a Bedtime Story' is the fourth result here. Radiohead's brit rock classic 'Karma Police' is slow, minor key song with a bittersweet character, a guitar and piano accompaniment, with prominent cymbal hits in the mix. Out-of-genre search results for this track include a pop song, Robbie Williams' 'She's the One', and a classic punk track, 'London Calling' by The Clash: both of these, however, share some obvious musical characteristics with the query. The remaining unexpected results, on the other hand, are plainly poor, such as a Mozart mass movement returned for a track by Moby, or ABBA, Deep Purple or the death metal band Sepultura to match Sonic Youth's experimental noise rock.

We see that in this setting the scale factor for musword counts serves effectively as a system parameter, controlling the influence of the audio content analysis on search results. Indeed one possibility in a practical search system would be to allow the user to vary this parameter at search time, controlling the balance between audio-based music discovery, with its increased risk of inexplicable 'clunkers', and purely word-based search with its tendency to recommend the obvious. We observe that the tracks in our test set are reasonably well-tagged. A further consideration in searching large collections is the effect of scaling musword counts in the presence of a large number of sparsely-tagged tracks. We investigate this in the following Sections.

It is possible to avoid the issue of scaling counts altogether by using more sophisticated models, such as an extended version of the aspect model, in which words and muswords are treated as being generated independently for each track. This has its own problems, however, most significantly a mismatch between our observations and the model structure. The underlying co-occurrence data for such a three-way model is a set of <*track, word, musword*> triples; in reality we do not know the association of individual muswords for a track with any of the particular words describing it. For this and other reasons this approach, while attractive, remains outside the scope of the present study.

4.2 An evaluation framework for joint models

For evaluation to be realistic, retrieval tasks have to be set in a scenario in which tracks for some artists are sparsely-tagged, as discussed in Chapter 3. This can be simulated in a cross-validation framework as follows:

1. the test set artists are split into three folds at random
   For each fold in turn:

2. the tag words for each track by the artists in the current fold are sorted by their count

3. all but the top κ words for each track are masked by setting their counts to zero

4. query by example is evaluated as before for all tracks in the test set

The three-fold harness both allows cross-validation and reproduces approximately the distribution of tags which we observed in the full dataset in Chapter 3: it simulates the scenario in which tracks by a third of all artists have been tagged with only some small number κ of words. A possible consequence of the uneven distribution of tags is that search results may effectively segregate tracks by sparsely- and well-tagged artists. Besides means and standard errors for genre and artist retrieval precision over the three folds, we therefore report a measure of track *integration*: the proportion of masked tracks appearing in the top ten search results for unmasked tracks, and vice versa.

4.3 The effect of tag sparsity

Figures 4.2, 4.3 and 4.4 show cross-validation results for query by example on the test set using the BOW+M representation in a Vector Space model with tf-idf in the framework described in the previous Section. The plots show how search results are affected by tag sparsity, and how they vary as we use words only (scale factor = 0), words plus muswords with counts scaled to have the same mean (scale factor = 1), and words plus muswords scaled to have more influence (scale factor = 2). The x-axis shows the number of words remaining after masking to simulate sparse tagging, i.e. all but the indicated number of top tag words are masked for tracks by the artists in each fold. The rightmost value of each corresponds to using all tags words for each track i.e. it shows a performance in the ideal scenario where tag sparsity is not an issue.

We can draw several conclusions from these results. Firstly, Figures 4.2 and 4.3 show that tracks remain highly organised in a BOW+M model even when tags are scarce: although it helps to have many words for each track, retrieval remains at state-of-the-art levels as long as we have more than one word for each track. Even with only a single word available for a third of our test tracks, performance far exceeds content-based methods, such as the baseline method shown in Table 3.1. This shows the 'wisdom of crowds' in action: by inspection the most frequently applied word in tags for a track is usually an appropriate genre label. Secondly, incorporating muswords into the model can actually increase retrieval performance when only a single word is available for a third of the tracks, as long as the counts are scaled appropriately. In particular artist organisation increases significantly when we introduce muswords, taking advantage of the so-called 'album effect', i.e. the ability of content-based representations to match highly similar tracks. Finally, we see from Figure 4.4 that

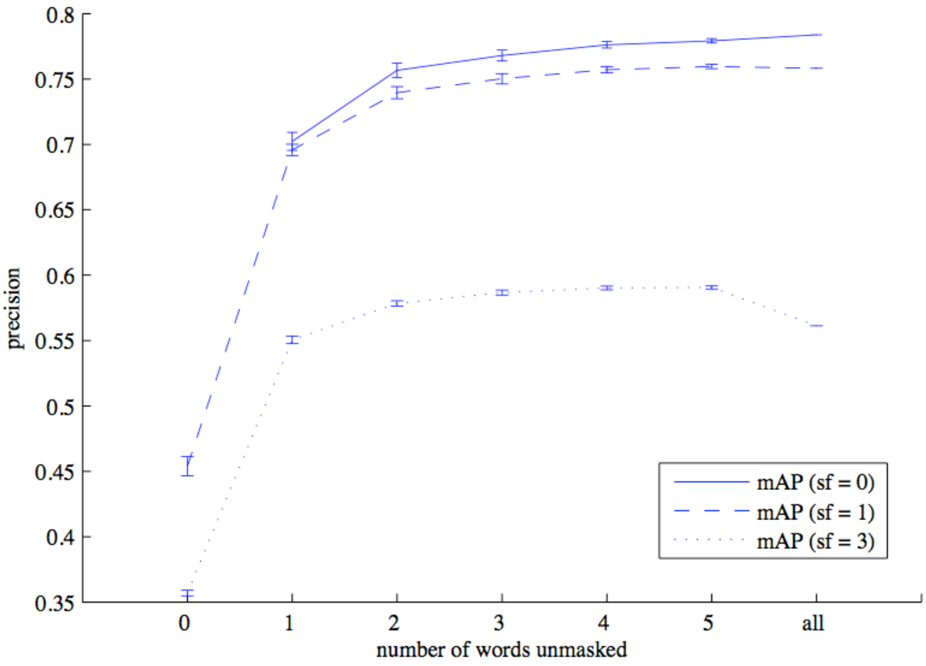


Figure 4.2: BOW+M genre retrieval performance with sparse tags

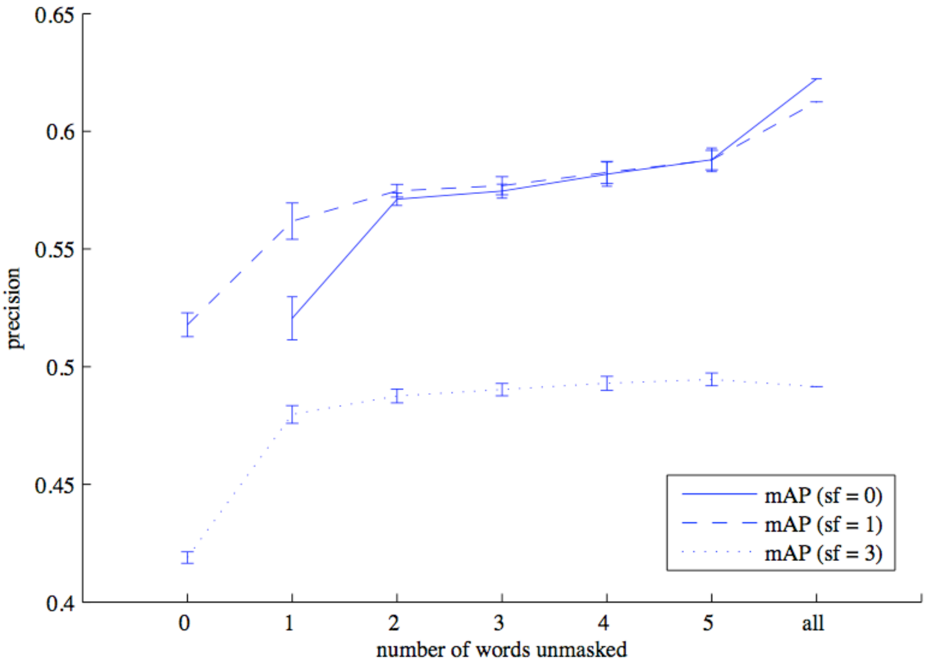


Figure 4.3: BOW+M artist retrieval performance with sparse tags

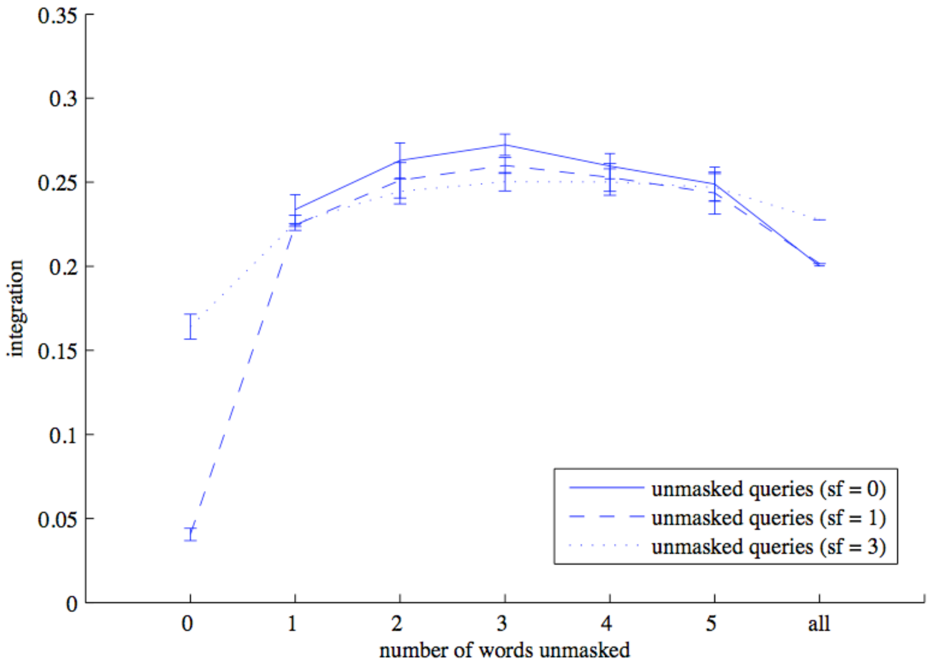


Figure 4.4: BOW+M integration with sparse tags

tag sparsity does cause some segregation in the Vector Space model. In particular we observe that on average there is less than one well-tagged track in the top ten search results for query tracks tagged with only a single word. Using muswords moderates this effect, but only makes a large difference if the scale factor for musword counts is high enough to degrade overall track organisation in the model.

These suggest that while current audio content-based information offers only limited help in solving the full cold start problem, i.e. with completely untagged tracks, it is useful in the context of sparse tagging. Specifically these results motivate the development of models trained on a joint vocabulary of tag words and audio muswords. In the following Sections we develop and evalute an aspect model of this kind, with word counts scaled so that the mean counts for conventional words and muswords are the same, and in the following Chapter we evaluate it as the basis of a practical system for query by keyword and automatic annotation.

4.4 Training an aspect model on words and muswords

One straightforward approach to training an aspect model on words and muswords is simply to apply the existing model of Chapter 2 to the counts over the joint vocabulary established in the preceding Sections. In Chapter 2, however, we saw that the aspects learned by models trained on conventional words alone were semantically coherent: high probability words for a given aspect clearly related to a common domain concept, such as a genre, era, nationality, particular artist, etc. Given the relatively poor correlation between current audio features and such domain concepts, this motivates an alternative two-stage training method, as suggested in [Monay & Gatica-Perez, 2007] where aspect models are applied to image annotation. In this two-stage training, semantic aspects are first learned by training on words only; the $P(z|t)$ for the training tracks are then held fixed during a further set of E-M iterations in which the $P(m|z)$ are learned for the muswords. Finally the word and musword probabilities $P(w|z)$, $P(m|z)$ are weighted by the total word and musword counts respectively, and normalised to sum to unity. The second stage of training is given in Algorithm 4.1, where the input probabilities are the output of Algorithm 2.3 shown earlier in Section 2.6. This two-stage training ensures that the aspects remain semantically coherent, while further tracks, particularly those that are sparsely- or un-tagged, can be folded in to the model using both words and muswords. In the following Sections we compare the retrieval performance of models trained by the simple and two-stage strategies.

Algorithm 4.1: Second stage training for a joint aspect model

```
Input: Probabilities $P(w|z)$ for $w$ in words, $P(z|t)$, total counts $n_{word}$,
       $c_{musword}$ , number of aspects $K$, musword vocabulary size $D$,
       training and validation sets of tracks, early-stopping threshold $\tau$
Output: Updated $P(w|z)$, probabilities $P(m|z)$ for $m$ in muswords
Initialise $P(m|z)$ to random values for $m$ in muswords
Initialise accumulators $W[D][K]$ to 0
Compute $L$ by folding in validation set
while increase in $L > \tau$ do
    $W[D][K] \longleftarrow 0$
    foreach Track t in training set do
        foreach Musword $m$ do
            foreach Aspect $z$ do
                $q[z] \longleftarrow P(m|z) * P(z|t)$
            end
            Normalise $q[z]$ to unit sum
            foreach Aspect $z$ do
                $W[m][z] \longleftarrow W[m][z] + c(t, m) * q[z]$
            end
        end
    end
    foreach Aspect $z$ do
        foreach Musword $m$ do
            $P(m|z) \longleftarrow W[m][z]$
        end
        Normalise $P(m|z)$ to unit sum over $m$
    end
    Compute $L$ by folding in validation set
end
Normalise $P(m|z)$ to sum to $c_{musword}$ over $m$ in muswords
Normalise $P(w|z)$ to sum to $n_{word}$ over $w$ in words
Append $P(m|z)$ to $P(w|z)$ and normalise to unit sum
```

4.5 Evaluation

Aspect models with a range of numbers of aspects were trained jointly on words and muswords, using both the one- and two-stage training strategies, and evaluated in the three-fold framework introduced in Section 4.2. The models were trained on the artist-disjoint training set of 5064 well-tagged tracks **ADW**. Audio was available for 2824 of these tracks; all available words and muswords for each training track were used in training, scaling musword counts to have the same mean as word counts. For each fold of the test set **Ta** either all or all but one of the tag word counts for the relevant tracks were masked before folding in the whole test set.

4.6 Results

The retrieval results given in Figures 4.5 and 4.6 show that aspect models trained by conventional E-M over the joint vocabulary perform poorly. Two-stage training, on the other hand, where we learn the aspects themselves from tag words only, gives retrieval performance only slightly below that of the vector space model, while solving the segregation of well- and sparsely-tagged tracks, as illustrated by Figures 4.7 and 4.8. For clarity the plots show mean AP only. The best genre precision at 5 for the two-stage model was 0.86, while the best artist r-precision was 0.44.

We observe further that we achieve these results despite adopting the extreme scenario in which none of our test artists were present in the training set. While this scenario gives us confidence that our models have indeed learned some semantics, in a practical application it can be avoided by a variety of means including training on the whole dataset if computational resources permit, representative subsampling of tracks, vocabulary pruning or incremental training with the use of approximate direct parameter updates if necessary. We find that retrieval performance with aspect models equals or exceeds that of the vector space model when the training set does indeed include tracks by test artists.

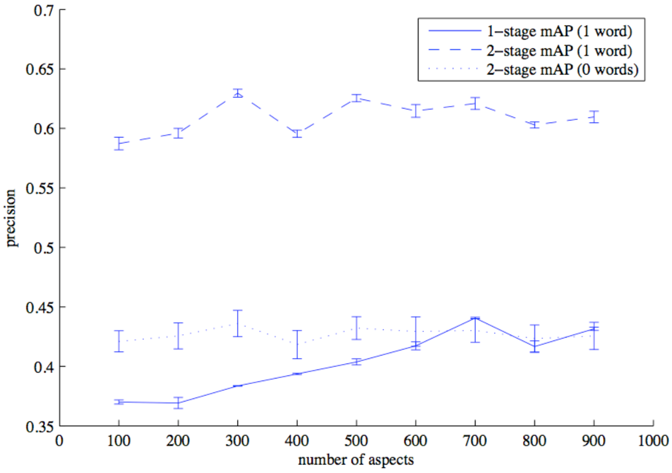


Figure 4.5: Aspect model genre retrieval performance with sparse tags

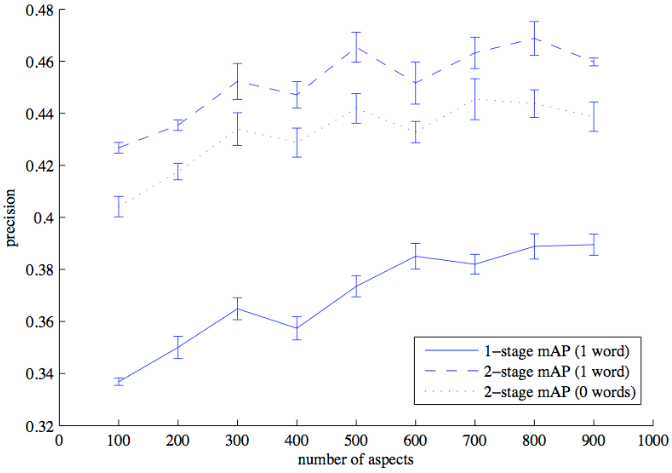


Figure 4.6: Aspect model artist retrieval performance with sparse tags

4.7 Conclusions

In this Chapter we indexed a joint vocabulary of conventional words, drawn from social tags, and muswords with vector space and probabilistic aspect models, and demonstrated how a scaling factor for word counts serves as a system parameter controlling the influence of audio over retrieval results. We

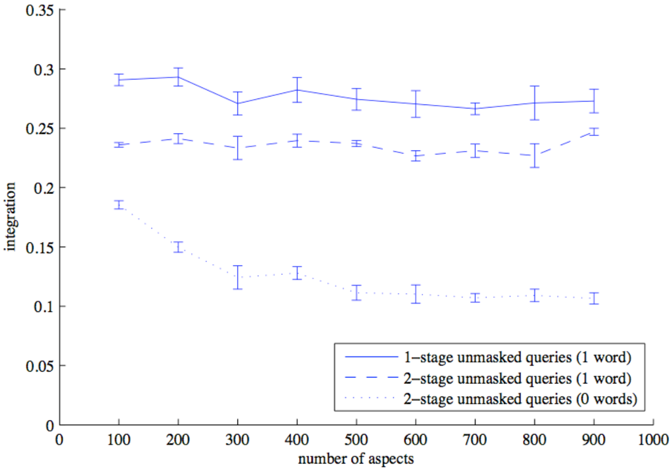


Figure 4.7: Aspect model integration with sparse tags: well-tagged queries

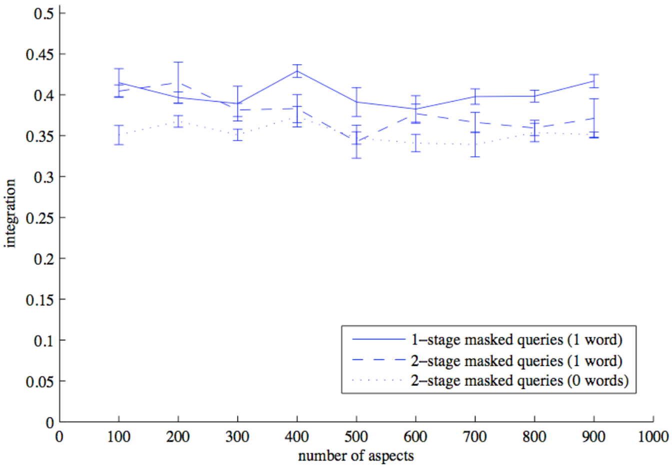


Figure 4.8: Aspect model integration with sparse tags: sparsely-tagged queries

saw how these models provide effective retrieval even under realistic conditions of tag sparsity: in particular retrieval is is excellent as long as two or more tags are available for each track, with the inclusion of audio making no significant difference to the performance in such cases. Retrieval is improved by indexing audio when fewer tags are available, as is the case in current real-world tagging systems, and indexing audio also helps to avoid segregation between sparsely and well-tagged tracks.

Social tags for music are increasingly being used in research, principally as a direct groundtruth for classification and retrieval tasks [Eck *et al.*, 2008; Knees *et al.*, 2007; Geleijnse *et al.*, 2007]. Most existing studies acknowledge, however, that real tags for music are in fact far from being idealised class labels, leading to a need to "normalise away" the subjectivity and informality that in fact typify social tags for music. The methods of this Chapter, on the other hand, outline an approach that can make good use of tags for music as they really are. The next Chapter builds on this, leading to practical systems for automatic annotation and semantic retrieval.

Chapter 5

Retrieval and annotation using semantic models

So far in this thesis, the evaluation of semantic models for music has centred on query by example, i.e. experiments in which the models are used to retrieve other tracks similar to a given query track. The use of a query by example scenario is motivated by two important practical applications: track or artist-based playlist generation and (internet) radio streaming. In both cases a music service is required to select a number - in the case of streaming, often a large number - of tracks similar to an initial seed track or artist specified by the user. Query by example is also an attractive scenario to use for evaluation because it makes it possible to verify the organisation of tracks according to a model against a credible groundtruth: tracks by the same artist and in the same genre as the query should come high up in the results.

In this Chapter, the models developed earlier are finally applied to the more challenging scenarios that motivated this research in the first place: semantic retrieval, i.e. query by keyword or free text, and automatic annotation of sparsely- or un-tagged tracks. Practical applications of semantic retrieval include playlist generation, radio streaming or, equivalently, catalogue search, given a keyword or free text query supplied by a user. A few current real world systems address these tasks. The All Music Guide[1] supports catalogue query by keyword, where the keyword can be drawn from a large vocabulary of specialist descriptive terms encompassing so-called *moods* and *themes* as well as more familiar labels for genre, instrumentation and nationality. Tracks are annotated by hand against a checklist of the terms, leading to obvious issues of scalability. Last.fm "tag radio" provides internet streams of tracks sharing a particular tag chosen by the user. This is scalable, and queries are in principle not constrained to a fixed vocabulary, provided that users as a whole continue to supply tags in large numbers. The variety of tracks chosen for these streams suffers, however, from the large number of artists whose tracks are at best sparsely-tagged, as discussed at the start of Chapter 5. This chapter explores the value of semantic models in relation to these issues.

Useful practical applications of automatic annotation for its own sake are harder to find. Perhaps the most intriguing is the prospect of a reliable music description machine, with futuristic consequences such as the computer-generated music reviews suggested in [Whitman & Ellis, 2004]. For the time being, a reasonable view of automatic annotation is as an intermediate step on the way to semantic retrieval, supplying descriptions that allow unannotated documents to be retrieved as easily as annotated ones. This seems particularly true for music, where weak semantics mean that associations between descriptions and tracks are better described with continuous relevance scores or probabilities than considered simply 'right' or 'wrong'. The task of annotating a track therefore corresponds to assigning suitable scores to each word in the vocabulary. Although a hard annotation can then be derived from these scores, for example by outputting some arbitrary number of highest scoring words [Turnbull *et al.*, 2008], output of this kind is very hard to evaluate directly, for example because there is no sensible figure for the "correct" number of annotations for any particular track.

[1] `http://www.allmusic.com`

If we accept that the primary purpose of machine annotation is to support semantic retrieval, then it is more sensible simply to evaluate retrieval performance and treat this as an implicit guide to the quality of annotation: in other words semantic retrieval and annotation reduce to a single task for evaluation purposes. Suppose for concreteness that we have scores according to a model for a collection of tracks for the word *slow*. Instead of attempting to measure the absolute relevance of this annotation to each track, we use the scores to rank the tracks by *slow*-ness, and then use well-established information retrieval measures to evalute the quality of the ranking. This argument has been largely accepted in the extensive parallel literature on automatic image annotation and retrieval [Monay & Gatica-Perez, 2007]. In this chapter performance statistics for automatic annotation are given for the sake of completeness, but the main evaluation focusses on semantic retrieval.

The remainder of this chapter is organised as follows: sections 5.1 and 5.2 explain how semantic models can be used respectively to supply annotations for sparsely-tagged tracks, and to improve retrieval of tracks matching semantic queries; section 5.3 describes an experimental setup for evaluation of automatic annotation, and of retrieval over a vocabulary of realistic queries; results, including examples of annotations produced by aspect models trained on words and muswords, and lists of tracks retrieved for semantic queries, are given in section 5.4, and conclusions are summarised in section 5.5.

5.1 Automatic annotation using aspect models

Given a trained aspect model, and a track t with aspect conditional probabilities $P(z|t)$, which we can obtain by folding in if t was not in the original training set, we can estimate the probability of each word w in the vocabulary being applied to t as follows:

$$P(w|t) = \sum_z P(w|z)P(z|t) \tag{5.1}$$

This "folding out" of the aspect probabilities can be seen as smoothing the probability mass associated with counts for each word observed in tags for a track across other words which we would expect to see given more observations i.e. more tags or other training annotations for t. In other words, $P(w|t)$ is a smoothed version of the empirical distribution $\hat{P}(w|t) = n(t, w)/n(t)$, which we obtain by back-projection from the latent semantic space into the original word space according to (5.1). So, for example, if $P(z|t)$ is large for a semantic aspect z relating to *motown*, as might happen if t were tagged with a highly characteristic word such as *motown* itself, then $P(oldies|t)$ and $P(60s|t)$ are likely to be significant according to (5.1), even if *oldies* and *60s* were not amongst the tags actually applied to t.

Using the joint models discussed in the previous chapter, aspect probabilities $P(z|t)$ are estimated even for completely untagged tracks from the audio muswords associated with them. This illustrates an important property of this smoothing approach: a single model can generate improved annotations for tracks that already have tags, as well as purely automatic annotations for untagged tracks. This is a far better fit to the real-world availability of annotations than the pure prediction approach, typically using banks of classifiers, pursued in previous work [Turnbull *et al.*, 2006, 2008; Eck *et al.*, 2008], particularly given the poor state of the art for such classifiers and the relative ease of obtaining, say, a single relevant human annotation for any given track.

Given the smoothed $P(w|t)$, a hard annotation can be output most simply by ranking the vocabulary according to $P(w|t)$ and retaining some arbitrary number of the highest ranking words. Although this approach has been widely used in the parallel image literature, more sophisticated strategies for choosing which words to output are possible. These include (i) creating separate decision rules based on $P(w|t)$ for each word in the vocabulary, either by taking into account their prior probabilities $P(w)$ or by hand-tuning against a validation dataset to optimise the ratio of true to false positives; and (ii) using a suitable information measure to determine the optimal number of words to output for each track. In the experiments described below, however, the simple ranking strategy outputting a fixed number of words for each track is used, to allow comparison with previous work.

5.2 Semantic retrieval using aspect models

In the case of semantic retrieval, our aim is to retrieve tracks from a collection which best match the user's query q, where q is a bag of words such as "cool jazz vocals", "ironic gospel", "funky 70s disco", etc. Given a trained aspect model and a set of tracks with aspect conditional probabilities $P(z|t)$, obtained by folding in if necessary, two approaches to semantic retrieval are proposed in Hofmann's original paper [Hofmann, 1999b], based on cosine distance in the original word space and the latent semantic space respectively. In the first approach, the smoothing of (5.1) is applied to word counts for each track in the collection, and the cosine distance between q and the smoothed count vector $P(w|t)$ is used as the score for track t. In the second approach, the query q is first folded in to the model to estimate its aspect probabilities $P(z|q)$, as described in Section 2.6, and the cosine distance between $P(z|q)$ and $P(z|t)$ is then used as the score for track t. In Hofmann's experiments over four different collections of standard text documents, both methods performed well, the best method varying from one collection to another. In the experiments described below, the second approach is adopted, i.e. cosine distance in the low-dimensional semantic space is used as the similarity score.

5.2.1 Related work

Semantic retrieval differs from the binary search by tag which is currently implemented in real-world systems such as Last.fm. These systems typically expect queries to correspond directly to an existing tag g, returning tracks tagged g in order of popularity. In contrast, using a semantic model allows us to return tracks tagged with words that are similar in meaning rather than necessarily identical to those of the query, and gives a natural ordering by semantic similarity. In addition, models trained jointly on words and muswords offer a simple way to take advantage of audio similarity to allow the retrieval of sparsely-tagged tracks within a single system.

Recent academic work on web-based music retrieval by Knees, Phole, Schedl and Widmer [Knees *et al.*, 2007], however, does provide a useful baseline for the retrieval results presented in this chapter. Knees et al. build a vector space model based on web-mined text for a collection of tracks. Although the model only indexes words, a timbral similarity metric is employed to smooth word counts by weighted averaging over acoustically similar tracks. As discussed in section 2.3, web-mining text for large numbers of tracks suffers from huge vocabulary sizes, even compared with social tags, as irrelevant content is inevitably included in the text to be indexed, making dimension reduction of some kind essential. Knees et al. use timbral similarity indirectly to prune the word counts for each track, retaining only the words that discriminate most effectively between a group of timbrally neighbouring tracks and a group of distant ones. The vocabulary that remains after this track-specific pruning is clearly highly fitted to the training set, and external queries have to be folded in by a process of massive expansion. In the current implementation, queries are first submitted to Google, then the top 10 pages returned are downloaded, and all their text aggregated, before finally indexing against the model vocabulary. Knees et al. evaluate their model in a free text query scenario, using the most popular Last.fm tags as queries, and treating Last.fm tags for each track directly as a groundtruth, achieving a best r-precision of 0.264 over a set of 227 test queries including genre and other terms.

While not directly comparable, Turnbull reports per-word mean Average Precision of 0.390 for semantic retrieval averaged over a set of 174 queries and based on a bank of classifiers trained on audio only [Turnbull *et al.*, 2008]. Independent work by Law, Settles and Mitchell reported in [Law *et al.*, 2010] pursues research related to the methods described here and previously published in [Levy & Sandler, 2009]. Although Law et al. take a classification approach to predict tags from audio only, their classifier is trained on posterior probabilities for latent topics for each labelled example. The topics themselves are first learned by Latent Dirichlet Allocation [Blei *et al.*, 2003]; at query time tag probabilities are inferred from the predicted topic weights for an unlabelled audio query. Law et al. use training and test data acquired via the TagATune online annotation game [Law & von Ahn, 2009]. Annotation and retrieval over a test vocabulary of some 200 tags are evaluated using both this model and a baseline method in which a separate binary classifier is trained for each tag. Law et al. report mean Average Precision of around 0.3 on a retrieval task, and precision and recall both around 0.25 for annotation of unlabelled audio. They also report results from a separate human evaluation for a subset of their test queries. Perhaps as a result of having used groundtruth data acquired through collaborative gameplay, they find that human evaluation suggests that offline metrics appear to underestimate the performance of their algorithms: overall their topic-based method performs similarly to their simpler baseline classifiers.

5.3 Experimental setup

5.3.1 Dataset and model training

For the experiments described in this chapter we would ideally like a large training set of tracks for each of which we have audio and a full set of trustworthy human semantic annotations to use as a groundtruth. Even disregarding issues of subjectivity in annotation, such datasets are not currently open to the research community. The approach taken here is consequently a pragmatic one, using some simplifying assumptions that make it possible to define realistic tasks for evaluation on the data available:

1. a dataset of well-tagged tracks with audio available is selected

2. tags for some tracks are withheld to simulate realistic tag availability
3. query by keyword over the whole dataset is evaluated, treating the withheld tags as a groundtruth
4. automatic annotation of tracks that are untagged is evaluated, treating the withheld tags as a groundtruth

The following sections give more details of each step.

The chosen dataset consists of the 2,824 well-tagged tracks for which audio is available in set **ADW**, as described in Section 4.5. For simplicity, and to allow comparison with previous work, the tags are treated directly as a groundtruth for both annotation and retrieval: a word w is considered to be a correct annotation for track t if it occurs amongst tags applied to t. Similarly when searching for tracks matching a query q, a track t is considered to be a correct hit if each word in q occurs amongst tags applied to t.

In order to investigate the usefulness of semantic models under realistic conditions of tag availability, tag words for some tracks are masked by setting their counts to zero, following a similar (though not identical) procedure to the one used in the experiments of Section 4.2. In this case a target distribution for the number of distinct words per track is first chosen, to simulate approximately the distribution of tags for each artist observed in our full set of tracks. The observed artist-wise tag distribution was illustrated in Fig. 3.1; the target track-wise distribution used here is shown in Fig. 5.1. The appropriate number of tracks is then chosen at random for each bin of the target distribution, and finally words for each track are masked by setting counts to zero for all but the target number of most frequently applied words. Note in particular that 30% of the tracks are completely untagged after masking.

Aspect models of various ranks are then trained on all the tracks in this dataset, using the masked words and, following the two-step training algorithm described in 4.4, all muswords for each track. In contrast to the experiments reported in previous chapters, the decision was taken not to use separate

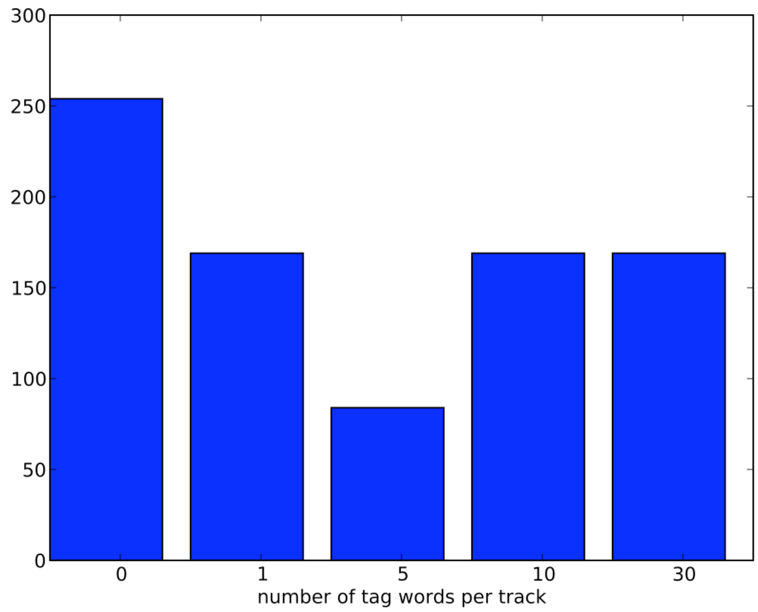


Figure 5.1: Masked tag distribution

training and test sets here, as the assumptions motivating this separation appeared unduly pessimistic for semantic retrieval in particular:

1. real-world retrieval systems aim to index the entire collection to be searched, i.e. in our context there is a very strong motivation to train models on as many tracks as possible, even if this means using parallel or approximate fast implementations, or making extra hardware available;

2. concerns over the scalability of this approach receded as with growing experience it became possible to optimise code to train aspect models using the standard E-M algorithm on hundreds of thousands of tracks in well under an hour on a single machine;

3. earlier experiments suggest good generalisation of aspect models even when many artists in a collection are completely unrepresented during training, i.e. retrieval performance even in the worst-case scenario is not likely to be much worse than in the best-case scenario adopted here.

5.3.2 Automatic Annotation

For each of the tracks in the dataset which are completely untagged after masking, automatic annotations are generated by outputting the ten top words according to $P(w|t)$ (5.1). Note that in contrast to automatic annotation using a bank of classifiers, the vocabulary of our automatic annotations is not constrained before annotation time (except by the overall vocabulary encountered in tags during training), and therefore varies from model to model. In practice models with more aspects tend to output a greater variety of words.

To avoid bias effects caused by the distribution of words in the groundtruth, and for easier comparison with related work, the annotations are evaluated by computing precision and recall for each word in the output vocabulary. The machine annotations always contain exactly 10 words for each track, while the groundtruth always contains more, frequently as many as 100 words, as illustrated in Figure 5.2, which shows the track-wise distribution of tags over the test tracks before masking. This means that there is an upper bound of less than 1 on the per-word recall possible with any annotation method, even one based on full knowledge of the groundtruth tags. A baseline method is used to estimate this upper bound for each word output by the model: this generates annotations by drawing 10 words at random from the groundtruth for each track. Note that the recall estimated from this baseline is only an approximation to a true upper bound for the performance of the model, due both to sampling effects and the fact that we evaluate recall over the words output by the machine algorithm rather than over a fixed vocabulary. Although per-track precision and recall avoid these issues, they can favour systems which output only the commonest words in the tag vocabulary, and per-word statistics have therefore been widely preferred in the parallel image annotation literature and in related work on music.

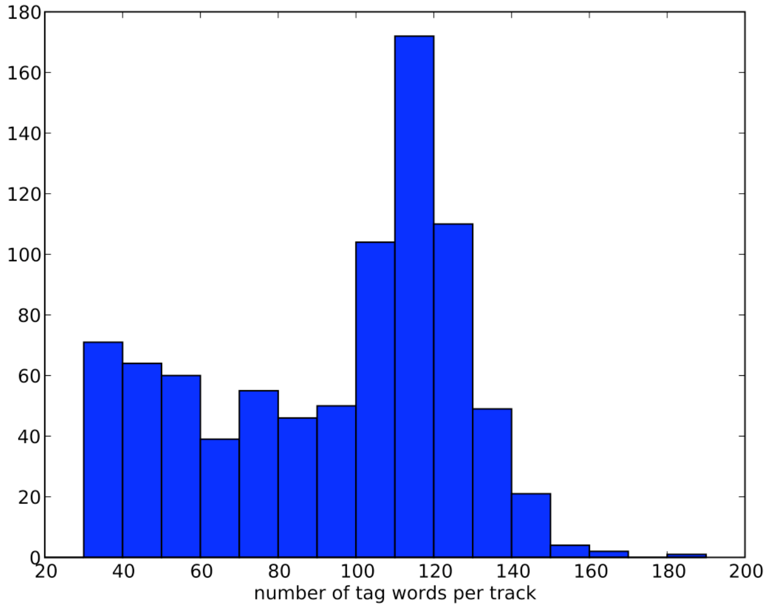


Figure 5.2: Groundtruth tag distribution

5.3.3 Semantic Retrieval

Following Knees et al., Last.fm's top tags at the time of writing[2] were used as a set of typical semantic queries: tracks are then retrieved from the masked dataset for each query. The complete list of queries is given in Table 5.1. A handful of the Last.fm top tags, describing user-track relationships rather than tracks themselves were, excluded from the list of queries: *favo(u)rite(s)*, *seen live*. As shown in Table 5.1, the remaining queries refer predominantly to genre, but also include era, nationality and mood.

For realism given the simple groundtruth defined here, retrieval is done in two stages: tracks whose available tags match the query directly are returned first, and a model is then used to find further tracks. More formally, given a trained aspect model, the retrieval algorithm is as follows: tracks containing all the words of the query q in their (masked) tag words are returned first; the query is then folded into the model, and the remaining tracks are ordered

[2] `http://www.last.fm.charts/toptags`, retrievd on 16 August 2008

Table 5.1: Semantic Queries

00s	60s	70s	80s	90s
acoustic	alternative	alternative metal	alternative rock	ambient
american	anime	atmospheric	avant garde	awesome
beautiful	black metal	blues	blues rock	british
britpop	brutal death metal	canadian	celtic	chill
chillout	christian	classic	classic rock	classical
comedy	cool	country	cover	dance
dark ambient	darkwave	death metal	disco	doom metal
downtempo	drum and bass	dub	easy listening	ebm
electro	electronic	electronica	emo	experimental
female	female vocalist	female vocalists	finnish	folk
folk metal	folk rock	french	fun	funk
german	goth	gothic	gothic metal	gothic rock
grindcore	grunge	guitar	hard rock	hardcore
heavy metal	hip hop	hiphop	house	idm
indie	indie pop	indie rock	industrial	industrial metal
instrumental	j pop	j rock	japanese	jazz
jpop	latin	lounge	love	male vocalists
melancholy	mellow	melodic death metal	metal	metalcore
minimal	new age	new wave	noise	nu metal
oldies	piano	polish	pop	pop punk
pop rock	post hardcore	post punk	post rock	power metal
progressive	progressive metal	progressive rock	psychedelic	psychedelic rock
psytrance	punk	punk rock	rap	reggae
rnb	rock	russian	sad	screamo
sexy	shoegaze	singer songwriter	ska	soul
soundtrack	stoner rock	swedish	symphonic metal	synthpop
techno	thrash metal	trance	trip hop	uk
viking metal	visual kei	world		

by their cosine distance from the query in the latent space. For evaluation purposes, the top r tracks are returned altogether, where r is the number of tracks containing all query words in their groundtruth (unmasked) tags.

As a baseline, tracks are also retrieved for each query following the same two-stage procedure but using a simple Vector Space model to rank tracks once all the exact matches have been found. This can return tracks matching some but not all of the query words. Finally, if r tracks have not yet been found, further tracks are returned simply in order of their overall number of tags, until r tracks altogether are again returned for evaluation. The results for each query are evaluated with r-precision, i.e. the precision (or equivalently the recall) at rank r. The r-precision is chosen in preference to mean Average Precision for this experiment because the algorithms being compared will return exactly the same tracks at low ranks (the tracks whose masked tags still contain all the words of the query).

Note that in contrast to the experiments described in previous Chapters, we have no trustworthy external groundtruth for general semantic retrieval. If such a groundtruth were available, i.e. a separate set of reliable annotations, and not the same set of tags which form the basis for retrieval, it might well be preferable to retrieve all tracks in a single stage using the model. In practice this would allow similarities learned from the overall distribution of tags for each track to override noise or poor annotation at the level of individual tags. In the absence of external annotations, however, deciding not to return a track tagged with all query words will always reduce the r-precision, whether or not the tags are truly appropriate for the track in question. The two-stage retrieval method described above was therefore adopted as a sensible way to evaluate the usefulness of the model given the unavoidable limitations of the experimental setup.

5.4 Results

5.4.1 Automatic annotation

Per-word precision and recall for 10-word annotations generated by a range of aspect models of different ranks are shown in Table 5.2, along with recall for the upper bound algorithm which draws words directly from the groundtruth, computed over the vocabulary output by each model. Table 5.2 also shows the total number of distinct words output in each case, as well as the number output with non-zero recall: remaining output words were not in the groundtruth for any track in the test set and are not evaluated.

Models with fewer aspects output a smaller vocabulary and have correspondingly higher precision and recall. Annotation with a model with 20 aspects is broadly comparable with the classification approach reported by Turnbull in [Turnbull *et al.*, 2008], which achieved per-word precision of 0.265 and recall 0.158 over a vocabulary of 174 concepts, of which 166 were output correctly. The different nature of Turnbull's training data, however, means that his estimated upper bound for recall of 0.375 is roughly three times higher than those estimated here for our dataset. In his setup a vocabulary is fixed in advance, his annotation dataset of questionnaire answers is constrained to stay within it, and each track is guaranteed to have roughly the same number of groundtruth annotations. In our case, there are simply more, and more varied, words applied to many tracks in the test set.

The vocabulary output by a 20-aspect model is given in Table 5.3, with words output correctly for at least one track shown in bold, and some example annotations output by a 100-aspect model are given in Table 5.4, illustrating the high proportion of relevant words output for some tracks in the test set.

5.4.2 Semantic retrieval

Table 5.5 gives the mean r-precision over the set of test semantic queries listed in Table 5.1 for the baseline method and for a range of aspect models of differ-

Table 5.2: Auto-annotation performance

aspects	precision	recall	words	output
		(upper bound)	output	correctly
500	0.107	0.025 (0.112)	747	676
100	0.174	0.052 (0.118)	390	372
20	0.260	0.113 (0.121)	135	132
10	0.302	0.176 (0.135)	77	74

Table 5.3: Machine annotation vocabulary for 20-aspect model

00	001	007	**00s**	01
010	011	**60s**	**70s**	**80s**
90s	**acid**	**acoustic**	**alternative**	**am**
ambient	**artists**	**avant**	**bass**	beat
beatles	**blues**	**british**	**britpop**	chanson
chill	**chillout**	**classic**	**classical**	**coast**
cool	**country**	**cover**	**covers**	**dance**
deutsch	**downtempo**	**drum**	easy	**electro**
electronic	**electronica**	**epic**	**experimental**	**favorite**
favorites	**favourite**	**favourites**	**female**	**folk**
francaise	french	**funk**	**fusion**	**garage**
garde	german	**girl**	**glam**	**grunge**
guitar	**hard**	hardcore	**heavy**	**here**
hip	**hiphop**	**hop**	hot	**house**
i	**idm**	**indie**	**instrumental**	**irish**
jazz	latin	listening	**live**	**lounge**
love	**male**	malloy2000	**melancholic**	**mellow**
metal	**motown**	**music**	**my**	n
neo	**new**	nu	**oldies**	**party**
peppers	**piano**	**playlist**	**political**	**pop**
post	**progressive**	**psychedelic**	**punk**	**queen**
rap	**red**	**reggae**	remix	**rnb**
rock	roll	**singer**	**ska**	**soft**
songs	**songwriter**	**soul**	**soundtrack**	spanish
stone	stoner	**techno**	**top**	**tracks**
trance	**trip**	turntablism	**underground**	**vocal**
vocalist	**vocalists**	**wave**	**world**	**york**

Table 5.4: Some machine annotations

Katie Melua: The Closest Thing to Crazy		Red Hot Chili Peppers: Under The Bridge		Jason Mraz: Tonight, Not Again	
female	0.159	rock	0.186	singer	0.082
vocalists	0.113	hop	0.071	songwriter	0.076
alternative	0.096	hip	0.060	rock	0.070
singer	0.043	funk	0.053	acoustic	0.060
songwriter	0.043	alternative	0.038	folk	0.050
soul	0.028	rap	0.025	mellow	0.044
jazz	0.027	cover	0.019	soft	0.027
piano	0.025	covers	0.018	artists	0.019
blues	0.021	classic	0.014	male	0.016
top	0.014	hard	0.013	jazz	0.015

Table 5.5: Semantic retrieval performance

model	mean r-precision
vector space	0.426
10 aspect	0.445
10 aspect + muswords	0.470
20 aspect	0.441
20 aspect + muswords	0.466
100 aspect	**0.480**
100 aspect + muswords	**0.531**
500 aspect	0.476
500 aspect + muswords	0.531

ent sizes. Results are given for aspect models trained on words only, as well as jointly on words and muswords, to separate out any possible benefit of using audio information from the effects of the semantic representation. Figure 5.3 shows the r-precision for each query in the test set for the baseline method and the best-performing aspect model: in each case the queries have been arranged in descending order of r-precision, to show how performance varies from word to word within the query vocabulary. The results show clearly that using the latent semantic representation provided by aspect models improves retrieval for queries at all levels of difficulty, while incorporating audio information improves it further still, with an overall improvement in average r-precision of 25% over the baseline.

Table 5.6 compares the top 20 tracks retrieved for the query *gothic rock* by

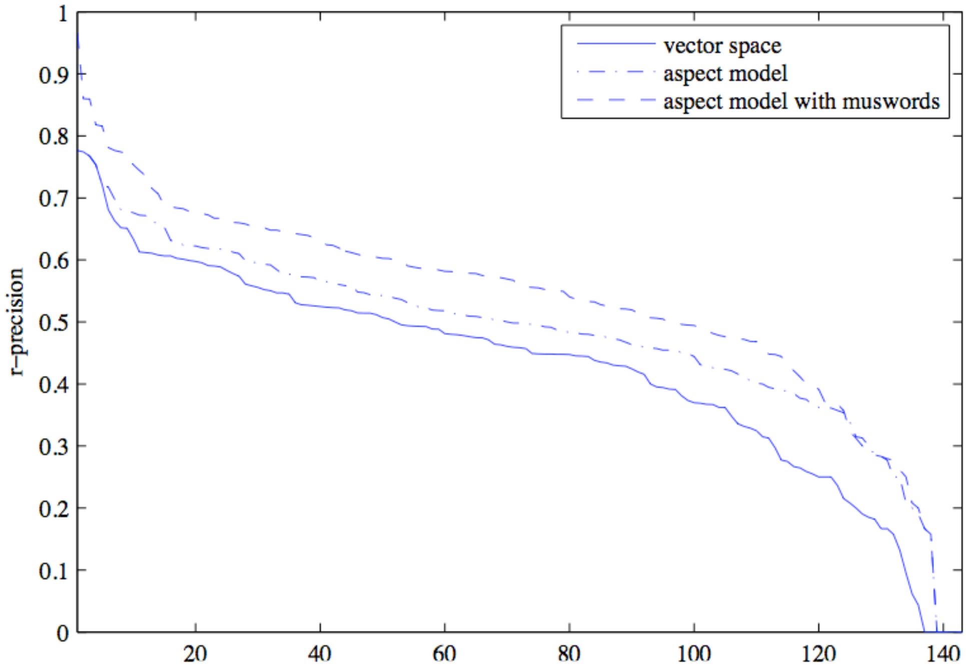


Figure 5.3: Semantic retrieval performance

the baseline method with those retrieved using a 100-aspect model trained on words and muswords. Tracks whose groundtruth tags do indeed contain both query words are shown in bold. The baseline method performs poorly for this query, because, after masking to simulate real-world tag sparsity, few tracks in the dataset are tagged *gothic*. The model easily overcomes this issue, although the presence of unexpected tracks marked as correct, such as a laid-back acoustic number by the singer-songwriter Jack Johnson, is a reminder of the shortcomings of our experimental setup described in Section 5.3.3: treating all tags directly as a groundtruth is clearly unrealistic.

To get some insight into the comparative performance of the methods given a stricter groundtruth, we can repeat the evaluation, but with a threshold on the count required for each query word for a track to be accepted into the groundtruth: tracks are only accepted as correct hits for a query q if every word w in q has been applied to the track at least θ times. As described in Section 2.2, the counts available from the Last.fm web service have been nor-

Table 5.6: Top hits for *gothic rock*

vector space	aspect model
The Velvet Underground: I'll Be Your Mirror	**Evanescence: Tourniquet**
Filter: Hey Man, Nice Shot	**Collide: Wings of Steel**
David Bowie: Speed of Life	**Nightwish: Dark Chest of Wonders**
John Lennon: Watching The Wheels	**Farin Urlaub: Sumisu**
The Rolling Stones: Sympathy for the Devil	***Queens Of The Stone Age: Someone's in the Wolf***
Sting: Englishman in New York	***Creedence Clearwater Revival: Bad Moon Rising***
Liquido: Narcotic	***Queens Of The Stone Age: Tangled Up In Plaid***
Pink Floyd: Another Brick in the Wall, Part 2	***The Smashing Pumpkins: Bullet With Butterfly Wings***
The Velvet Underground: Sunday Morning	**Tocotronic: Hi Freaks**
The Beatles: The Ballad of John and Yoko	***Linkin Park: Nobody's Listening***
The Beatles: Got to Get You into My Life	***Jack Johnson: Fortunate Fool***
The Verve: Lucky Man	**Nine Inch Nails: And All That Could Have Been**
Pink Floyd: The Fletcher Memorial Home	**Apocalyptica: Kaamos**
Electric Light Orchestra: Mr. Blue Sky	***Marilyn Manson: The KKK Took My Baby Away***
Pink Floyd: Us and Them	***Opeth: Death Whispered a Lullaby***
R.E.M.: Everybody Hurts	***Linkin Park: Figure.09***
The Verve: The Rolling People	***Nena: 99 Luftballons***
U2: Where The Streets Have No Name	The Beta Band: Push It Out
U2: Electrical Storm	The Verve: The Rolling People
Queens Of The Stone Age: First It Giveth	Tortoise: I Set My Face to the Hillside

malised so that the largest value for each track is 100, and rounded down so that relatively infrequent tags have an apparent frequency of zero: we increment them to give non-zero values. The threshold θ consequently specifies the frequency of w relative to the word most often applied in tags for the track in question, and by inspection a threshold of $\theta = 6$, i.e. accepting only words whose count is at least 5% of that for the top word for each track, eliminates most of the obviously poor examples from the groundtruth. Tracks removed from the groundtruth for *gothic rock* by this threshold are shown in italics in Table 5.6. Note that in general while less tracks will be accepted as correct hits against this groundtruth, the r-precision will not necessarily go down, because r is typically smaller for a given word and so less hits will be evaluated.

Retrieval results based on the thresholded groundtruth are given in table Table 5.7. Retrieval performance with aspect models trained jointly on words and muswords is virtually unaffected, in fact improving slightly for models with less aspects. Aspect models trained on words only, however, perform

Table 5.7: Semantic retrieval performance, thresholded groundtruth

model	mean r-precision
vector space	0.458
10 aspect	0.505
10 aspect + muswords	0.519
20 aspect	0.506
20 aspect + muswords	0.519
100 aspect	0.506
100 aspect + muswords	0.519
500 aspect	0.505
500 aspect + muswords	0.518

better against this groundtruth, and the baseline results are significantly better: the best-performing models nonetheless still show an improvement of 13% over the baseline.

5.5 Conclusions

In this Chapter aspect models trained jointly on words and muswords were used to annotate completely untagged tracks, and to do semantic retrieval from a set of partially-tagged tracks. To create a realistic test set for semantic retrieval, annotations were masked to simulate the real-world availability of tags: in particular 30% of the tracks in this set were competely unannotated after masking.

Annotation performance is comparable with a state of the art classification approach (despite the more challenging nature of the data used here), while retrieval performance is roughly twice as good as that of the most similar system reported in the literature. Retrieval performance using aspect models trained on words improved over a baseline method for virtually all queries, with training jointly on audio muswords improving results still further. During evaluation it became clear that, in contrast to earlier experiments based on query by example, noise in the tags is an issue for semantic retrieval. Retrieval performance did not suffer when evaluated against a stricter groundtruth, but this does raise the possibility that better results could be achieved by the models if infrequently applied tags had been removed before training. A full investigation of this remains for future work.

Chapter 6

Emotion aspects of the semantic space of music

In preceding Chapters we have seen how dimension-reduction methods such as LSA and its probabilistic equivalents can be applied to social tags for music, both to build practical solutions for information retrieval tasks, and directly to reveal semantic aspects of the space of descriptions of music. The subjective coherence of the aspects learned by the models (as illustrated in Section 2.9, together with the quantifiable success of the models in practical tasks (as reported in Sections 2.8, 4.6 and 5.4), suggests that the aspects can reasonably be seen as a set of meanings informing the way in which listeners choose to associate particular words with individual tracks. While of course this leap from so-called "semantic" models to meanings understood by real people remains only an audacious hypothesis, it does suggest that it ought to be possible to apply empirical methods to large numbers of social tags to produce convincing results of interest to the broader study of music: this Chapter attempts to do exactly that.

As discussed in the introduction to social tags for music presented in Chapter 2, while the mechanism of social tagging is designed primarily to support classification and retrieval, the usage in the track-level tags considered here suggests that a much broader range of motivations is in play during the act of tagging music. Tags for tracks are frequently discursive rather than simple labels, they are often personal and spontaneous in nature, and they employ a very wide vocabulary: in particular they contain a wide range of words describing emotions. Inspired both by the frequent occurrence of emotion words in tags, and the striking similarity between the latent semantic spaces explored in Chapters 2-5 and the emotion spaces traditionally studied by psychologists (described below in Section 6.2), this Chapter presents an analysis of the use of emotion words in social tags for music, set in the context of the rich literature on emotional responses which already exists in the field of music psychology.

It would, of course, be naive to suppose that the act of tagging a track with an emotion word is equivalent to the responses, typically questionnaire answers, collected under controlled conditions in psychological experiments specifically designed to study listeners' experience of emotion in music. In general we know little about the identity of individual taggers, and we have no control over the circumstances in which they applied any particular tag. Indeed we cannot guarantee that they have even listened to the track in question, and we certainly cannot say whether an emotion word in some particular tag was intended to describe how the track made them feel, or what they thought the track was trying to express, or something else altogether (perhaps their girlfriend dumped them while they were listening).

On the other hand, tags have some enormous advantages compared with laboratory experiments as a means of collecting written emotional responses to music. Tags are supplied spontaneously by listeners in relation to music of their own choice and under normal listening conditions; the vocabulary used to describe emotion is not prescribed; and emotion annotations in tags are available in virtually unlimited quantities. This last advantage is particularly important. Laboratory experiments are designed to minimise the effect of biases and context effects in small samples. Given a sufficient volume of data, however, we can reasonably expect that such effects will simply cancel out or, at worst, add some small amount of noise to overall statistics in a large sample of tags. Indeed it is difficult to think of a realistic scenario in which context effects could cause systematic bias in the statistics of a large set of emotion words in tags: if more than a few couples split up while listening to a particular track then we can reasonably assume that some genuine emotional mechanism is at work here! It remains difficult to distinguish words intended to describe emotions that the listener actually experienced during listening from those listing emotions that they perceived to be expressed by the music: but this is a weakness of any method relying on self-reporting of emotion, rather than a particular shortcoming of tags in this context.

Like any other internet medium, tags are subject to deliberate spam, and at any given time some proportion of spam tags are likely to remain unfiltered by tagging systems, however hard their designers struggle to keep up with the behaviour of spammers. The well-known cases to date are largely playful in nature, for example the appearance of unexpected artists on Last.fm's *brutal death metal* tag page or radio stream, following mass tagging by users who surely knew that this tag is a poor description for artists such as Paris Hilton or Rick Astley. While emotion words do not seem an obvious target for spammers, in the work that follows simple measures are taken to mitigate the effect of unfiltered spam tags in the dataset used.

In this chapter we consider tags containing emotion words simply as unconstrained verbal responses to the tracks they describe, and look to the statistics of a large collection of such responses to show the extent to which the resulting associations are arbitrary, or whether they exhibit meaningful patterns. Specifically we apply semantic models to emotion words occurring in tags for tracks, and show how the resulting low-dimensional representations relate to traditional constructs in music psychology such as the circumplex [Russell, 1980] and the dimensional theory of affect [Posner *et al.*, 2005]. This Chapter also explores how emotion words relate to musical genre, and demonstrates how a joint mapping of tracks and emotion words might be used as the basis of novel interfaces to music collections. The remainder of this Chapter is organised as follows: Section 6.1 explains how a vocabulary of emotion words can be mined from tags; Section 6.2 compares a semantic model of this vocabulary to the classic circumplex of musical affect; Section 6.3 investigates how musical genres are characterised in this emotion vocabulary; finally Section 6.4 introduces the use of Correspondence Analysis to plot tracks and emotions in a joint space that can support psychologically-motivated browsing interfaces for music.

The approach described in this Chapter was first published in [Levy & Sandler, 2007] and subsequently developed further in collaboration with the music psychologist Gunter Kreutz, while he was a Research Fellow at the Royal Northern College of Music. Prof. Kreutz also helped set up the expert selection of emotion words reported in Section 6.1. Independent work extending the methods given in [Levy & Sandler, 2007] to a categorical model inspired by the theory of basic emotions was reported in [Laurier *et al.*, 2009b,a].

6.1 A vocabulary of emotion words in tags

The dataset of tracks studied in this Chapter (which was later extended to form the full dataset described in Chapter 3) was chosen to include a wide range of artists and also to ensure that a large number of emotion words were represented in their tags. Tags were retrieved for 8,872 tracks, including songs by some 2,700 artists from all the well-known popular genres, and a few classical pieces by the best-known 18th and 19th-century composers. All available tags for each chosen track were retrieved from the MyStrands and Last.fm web services. The MyStrands service provides all the tags ever applied to a given track, while the Last.fm service supplies up to 100 tags, ordered by the frequency with which users have applied them to the track in question. While the dataset clearly contains only a small subset of the total number of track-level tags available, it is still several orders of magnitude larger than the number of responses that can be collected in even a very large-scale laboratory experiment, containing over 330,000 individual annotations.

Information about the individual users who originally supplied the tags is not available from the web services, and while the Last.fm service gives "counts" indicating the frequency with which particular tags have been applied to a given track, as described in Section 2.2, these are relative values based on an unexplained normalisation, and are frequently zero. Put simply, we do not know which, or even exactly how many, listeners applied any particular word to a given track: an important consequence for the work in this chapter is that we therefore cannot know the extent to which some particular word is applied consistently by different listeners. To use the language of experimental psychology, we have no robust direct measure of *inter-rater agreement* for responses to a given musical *stimulus*. This means we cannot support assertions about the appropriateness of some particular emotion word to describe any specific individual track, in the manner of traditional music psychology. On the other hand, the size of the dataset, and the approach taken here, do ensure that we can make statements with confidence about the relationship of one emotion word to another, and of particular emotions to large numbers of tracks.

The choice of tracks for which to collect tags was seeded with both artists and emotion words. Tracks were first selected for a set of well-known artists balanced across the mainstream musical genres, based on the list described in Section 2.7. Further tracks were chosen by querying the web services for tracks tagged with with words in Hevner's seminal checklist of musical expression words, shown in Figure 6.1 (and discussed further below) [Hevner, 1935, 1936], which was expanded and updated to give a total of 366 words by adding all synonyms from WordNet [Fellbaum, 1998] for each word in the original checklist. Finally tags were collected for some 3,000 further tracks from an existing research collection. The scale of the aggregated dataset was chosen to give reasonable coverage across tracks and terms without becoming computationally intractable. Note that emotion words in the tags collected for each track are by no means restricted to those in the expanded checklist used to seed the selection of tracks. Conversely, a word in the checklist is not guaranteed to appear in tags for any of the tracks, if, for example, the word has fallen out of current usage to describe music: in this case the web services will simply return no results when queried for tracks tagged with the outmoded word in question.

A three-stage filtering process was used to establish a vocabulary of emotion words from these tags. Words applied to less than 50 different tracks in the dataset were first discarded, to avoid over-dependence on the particular tracks under consideration. The remaining 1,142 widely-used words were then inspected by hand, and reduced to a list of 174 candidate words which could plausibly refer to emotion. Finally these candidate words were presented to two expert raters: two experienced music psychologists (one male, one female) were given a forced-choice task to decide whether or not each of the terms was a meaningful description of an affect, emotion or mood that was appropriate to apply to music. Words judged by both experts to be appropriate were retained.

This resulted in a final vocabulary of 105 emotion words: *aggressive, angry, angst, atmospheric, bitter, bittersweet, bright, calm, cheerful, chill, chilling, comfort, contemplative, crazy, creepy, crying, cute, dark, deep, delicate, depressed, depressing, depressive, dirty, downbeat, downtempo, dreamy, driving, earnest, emotional, emotive, energetic, ethereal, exciting, feelgood, feeling, fiery, fun, funny, gentle, gloomy, happy, haunting, hypnotic, inspiring, intense, intimate, joy, joyous, light, longing, lush, majestic, meditative, melancholic, melancholy, mellow, merry, moody, mournful, moving, mystical, noir, nostalgic, passionate, peaceful, playful, poignant, positive, power, powerful, pure, quirky, reflective, relaxed, relaxing, romantic, rousing, sad, sadness, sensual, sentimental, serene, serious, sexy, sleepy, soaring, soothing, soulful, spiritual, sunny, sweet, sweetness, trance, tranquil, trippy, triumphant, uplifting, warm, weird, wistful, witty, wry, yearning.*

Expert selection was chosen here as a pragmatic method of establishing a vocabulary with reasonable confidence, although it clearly risks a mismatch between the expertise of trained music psychologists and the usage current amongst the tagging community. By inspection, however, only three of the 105 words may have been subject to misinterpretation, and even these words are likely to be used to refer to emotion in some cases: *trance* is most commonly used in tags as a genre label; *merry* can refer simply to the popular j-rock group of the same name; while *driving* is frequently used to identify tracks to listen to in the car.

6.1.1 Related work

Comprehensive lists of emotions expressed or evoked by music, commonly arranged into groups of semantically similar terms, have been important to psychologists since the pioneering work of Kate Hevner in the 1930s. The most direct use of such lists is in the design of questionnaires or other instruments intended to capture and categorise responses to music, in particular to support research into musical expression i.e. the relationship between technical characteristics of a particular piece of music or performance and the emotions experienced or identified while listening to it. Hevner's original papers considered musical expression in relation to major and minor modes [Hevner, 1935] and tempo and melodic structure [Hevner, 1937], and her checklist and experimental design continue to be influential to this day, with studies from the last few years including [Iwanaga, 1997; Schubert, 1999; Gabrielsson & Lindström, 2001; Collier, 2007].

The language of the 1930s can of course appear dated today and indeed several previous laboratory-style studies have updated Hevner's checklist to account for changes in usage [Farnsworth, 1954, 1969; Gabrielsson & Lindström, 2001]. The most recent update is Schubert's 2003 study [Schubert, 2003], in which 133 university music students were asked to rate the words in the checklist, along with an additional 23 words drawn from other sources [Russell, 1980; Whissell, 1989], for their suitability "for describing any kind of mu-

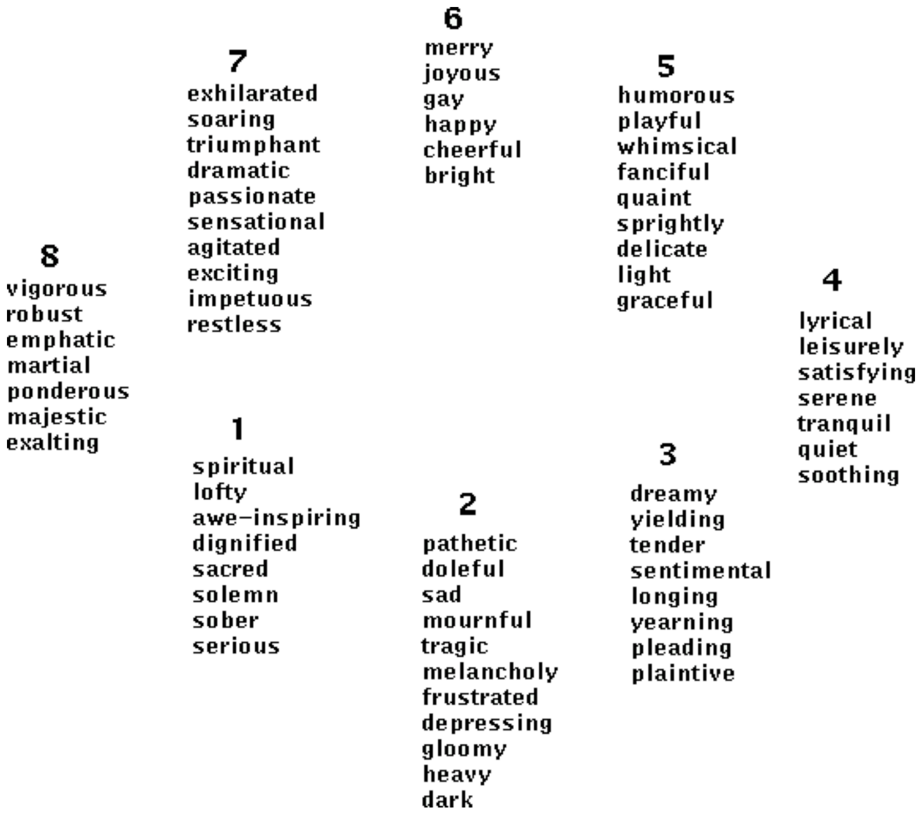


Figure 6.1: Hevner's checklist

Table 6.1: Schubert's updated version of Hevner's checklist

Cluster A	*bright, *cheerful, *happy, *joyous
Cluster B	humorous, *light, lyrical, *merry, *playful
Cluster C	*calm, *delicate, graceful, quiet, *relaxed, *serene, *soothing, tender, *tranquil
Cluster D	*dreamy, *sentimental
Cluster F	*dark, *depressing, *gloomy, *melancholy, *mournful, *sad, solemn
Cluster G	heavy, *majestic, sacred, *serious, *spiritual, vigorous
Cluster E	tragic, *yearning
Cluster I	agitated, *angry, restless, tense
Cluster H	dramatic, *exciting, exhilarated, *passionate, sensational, *soaring, *triumphant

sic". The resulting updated list, shown in Table 6.1, includes 41 of Hevner's 67 original words and just two of the new candidates: words also found in our tag emotion vocabulary are marked with an asterisk. As Table 6.1 illustrates, the tag vocabulary includes the great majority of Schubert's words, including some from each of his clusters of similar terms (note that the order of the clusters follows Schubert's paper, where the letters used as cluster names refer to an earlier update of Hevner's list).

While some words in the tag vocabulary could be considered substitutes for words in Schubert's list not commonly found in tags, or equivalent new coinages such as *chill* or *mellow*, the tag vocabulary clearly covers a larger emotional landscape. Despite the specific request to subjects in Schubert's experiment to rate the suitability of words to describe "*any kind* of music", the resulting list of words is somewhat chaste, getting at most *agitated* or *exhilarated* while tags can be *aggressive, feelgood, sensual, sexy* or downright *dirty*. Given the recent date of Schubert's study, this almost certainly reflects a degree of self-censorship, or at the very least an unconscious bias towards the conventional protocols of classical music, whether on the part of the experimenter in choosing candidate words, or in the ratings given by his young college student subjects: such biases are not present in the context of tagging, where viewers of a listener's tags can be presumed in general to be peers. The tag vocabulary also adds *haunting* and *hypnotic*, suggesting a further significant semantic cluster not present in Hevner's or Schubert's lists. More generally, we observe that the non-intrusive origin of the tag data - where music and vocabulary are both chosen freely by listeners - does indeed lead to a different set of emotion words for music from those collected in laboratory-style experiments.

6.2 Emotion tags and the Circumplex

6.2.1 The circumplex

Hevner's list of words, as presented in the circular arrangement of Figure 6.1, proved an immediate precursor of the so-called *circumplex model of affect* first proposed by Schlosberg [Schlosberg, 1941] and widely discussed ever since (see [Larsen & Diener, 1992; Plutchik & Conte, 1997; Remington *et al.*, 2000; Posner *et al.*, 2005] for some recent reviews of the literature). In the circumplex, emotions are positioned around the circumference of a circle in such a way that the distance between words in the model reflects their similarity i.e. neighbouring words on the circle are maximally similar while words on opposite sides of the circle are maximally dissimilar, typically being polar opposites such as *happy* versus *sad*. Figure 6.2 shows a recent example, where the emotion words are drawn from the domain of consumer product design.

As Figure 6.2 illustrates, the circumplex has been adopted across a huge range of psychological domains, and circumplex arrangements of emotions have been found by applying statistical techniques to a wide range of types of data, including self-reported affect, similarity judgements, responses to photographs of facial expressions, etc. [Remington *et al.*, 2000]. While the resulting arrangement of emotions is sometimes considered simply as expressing a set of independent bipolar relationships between basic emotions, the circumplex has increasingly been regarded as a particular instance of the more general *dimensional theory* of affect. A recent study by Posner, Russell and Peterson [Posner *et al.*, 2005] helpfully summarises the dimensional theory and its main rival, the theory of *basic emotions*. In contrast to most previous work, where it can sometimes be unclear to the non-specialist what exactly is being modelled by

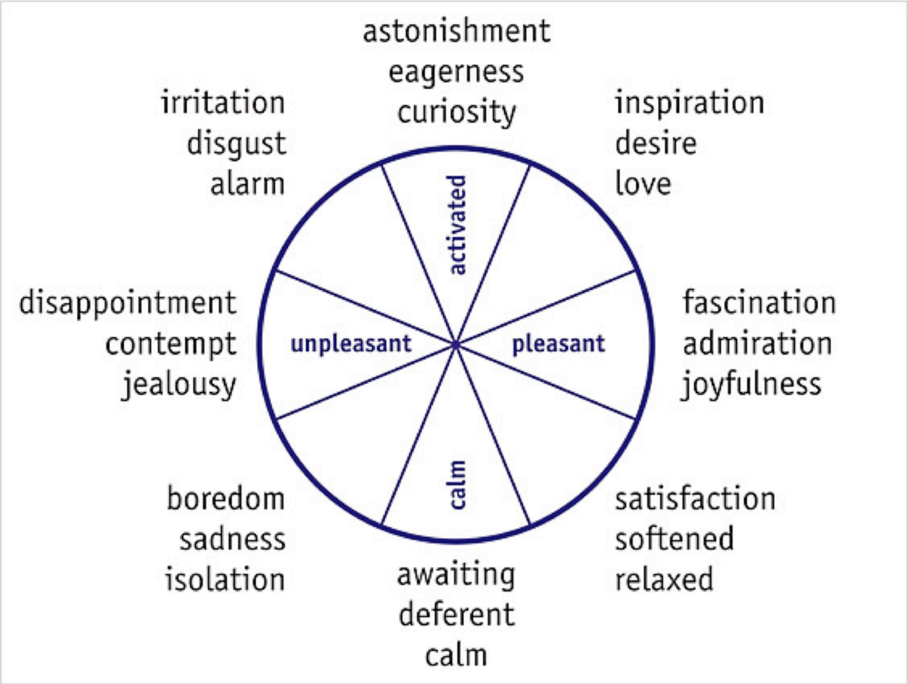


Figure 6.2: Circumplex model of affect
Copyright: ©2007 Desmet and Hekkert. Reproduced under Creative Commons license from [Desmet, 2008].

the circumplex, Posner et al. attempt to unite work from neuroscience and cognitive psychology, and introduce the circumplex as a simple model of the very neural systems which engender our experience of emotion.

Posner et al. describe the theory of basic emotions as follows:

> The dominant theory of emotion in psychiatric and neuroscience research posits that humans are evolutionarily endowed with a discrete and limited set of basic emotions... Each emotion is independent of the others in its behavioral, psychological, and physiological manifestations, and each arises from activation within unique neural pathways of the central nervous system... This is a theory in which each specific emotion maps to one neural system.

The dimensional model in contrast is based on recurrent observations of

> the difficulty that people have in assessing, discerning, and describing their own emotions... This difficulty suggests that individuals do not experience, or recognize, emotions as isolated, discrete entities, but that they rather recognize emotions as ambiguous and overlapping experiences... Dimensional models regard affective experiences as a continuum of highly interrelated and often ambiguous states.

The circumplex is identified directly with a two-dimensional model of affect:

> Although poorly represented in psychiatry, dimensional models have a long history in psychology... One particular dimensional approach, termed the circumplex model of affect, proposes that all affective states arise from two fundamental neurophysiological systems, one related to valence (a pleasure-displeasure continuum) and the other to arousal, or alertness... Each emotion can be understood as a linear combination of these two dimensions, or as varying degrees of both valence and arousal... Joy, for example, is conceptualized as an

> emotional state that is the product of strong activation in the neural systems associated with positive valence or pleasure together with moderate activation in the neural systems associated with arousal.

While not all psychologists accept that the circumplex directly models neural systems, there is widespread support for the view that mappings from disparate datasets producing similar circular arrangements of emotions show that the circumplex successfully conceptualises some essential property of human affective processing.

6.2.2 Modelling emotion words in tags

We can model the relationship of emotion words in our tag data by representing each word by its vector of track occurrences in the Vector Space model of Section 2.4, or, similarly, its dimensionally-reduced equivalent in one of the latent semantic models developed in Sections 2.5 and 2.6: instead of considering rows of the document-term matrix representing tracks we now simply consider columns representing words.

Formally, we represent word w by the vector $\mathbf{w} = [n(t_1, w), ..., n(t_m, w)]$ where t_j is the j-th track in our collection of m tracks, and $n(t_j, w)$ is the number of distinct tags applied to t_j which contain w. After applying semantic reduction to the document-term matrix $\mathbf{N}$ using LSA at rank k, or by training an aspect model with k aspects, word w is represented by a k-dimensional vector in the resulting semantic space $\hat{\mathbf{w}}$. The indivual elements of $\hat{\mathbf{w}}$ then represent the projection of $\mathbf{w}$ onto the k-th semantic axis in LSA, or the probability of w conditional on the k-th aspect $P(w|z = k)$ in the aspect model.

The similarity between two words w and w' can then be modelled by the cosine distance between their track vectors $\mathbf{w}$ and $\mathbf{w}'$

$$s(w, w') = \frac{\sum_t n(t, w)n(t, w')}{\sqrt{\sum_t n(t, w)^2}\sqrt{\sum_t n(t, w')^2}} \tag{6.1}$$

or their k-dimensional semantic equivalents, $\hat{\mathbf{w}}$ and $\hat{\mathbf{w}'}$. Note that the cosine distance is a similarity score i.e. it takes its maximum value when the vectors for w and w' are identical: to create mappings requiring an increasing distance measure we can use $d(w, w') = 1 - s(w, w')$ as the distance between words w and w'.

We observe that distances between emotion words computed in this way are robust to inter-rater inconsistency in tagging individual tracks, because they depend on the pattern of co-occurrence of words across a large number of tracks (8872 in the dataset used here). The distance between two emotion words in this representation changes significantly only when both words are applied to some non-trivial number of common tracks: while some individual co-occurrences of words and tracks in a large set of tags are likely to be spurious, co-occurrences across sets of tracks are much more likely to represent genuine associations. We note further that the design of the counts $n(t, w)$ used here is different from those in the models built in Chapter 2 to compare tracks: by neglecting the Last.fm "counts" here, and simply counting the number of distinct tags containing word w, the effect of spamming of any particular tag is minimised.

Various visualisation techniques can now be applied to the set of track vectors $\mathbf{W_E} = \{\mathbf{w} | \mathbf{w} \in E\}$ representing the words in our tag emotion vocabulary E, or to the matrix of pairwise distances $\mathbf{D_E} = \{d(w, w') | w, w' \in E\}$ between them, and the resulting mappings inspected to see if evidence of a circumplex arrangement, or any other low-dimensional semantic organisation, does indeed emerge from tag data.

6.2.3 Circumplex 2.0

To avoid over-interpretating artefacts of some particular visualisation technique, two quite different methods were used to create mappings, based on $\mathbf{W_E}$ and $\mathbf{D_E}$ respectively.

Self-Organising Maps

A simple two-dimensional mapping was first generated by training a Self-Organising Map (described previously in Section 3.2.1) on the full-rank track vectors for the emotion words $\mathbf{W_E}$, and mapping each word onto its vector's best-matching unit in the trained SOM. The map topology is a 10 x 10 rectangular grid, and a Gaussian neighbourhood function was again used during training. The resulting configuration of words is shown in Table 6.2. A relationship to traditional arousal-valence axes is immediately evident, with valence increasing clearly from bottom to top (*sad* to *happy*) and arousal generally from right to left (*relaxing* to *exciting*).

The mapping of Table 6.2 is rather congested in the lower left-hand corner, however, giving a poor idea of the larger-scale topology, perhaps because of the high dimensionality of the input vectors. We can get better discrimination between these terms by training on the SOM on a lower-dimensional representation of the emotion words, for example using LSA at rank 40, and learning a larger grid, as shown in Table 6.3. Here we see some sign of the conventional circumplex in the arrangement of words around the periphery of the mapping, particular in pairs of polar opposites such as *mystical, gentle, reflective* and *power, feelgood, driving* at centre top and bottom respectively, or *fiery, rousing exciting* and *dreamy, downtempo, relaxing* at centre left and right. We can also see the sequence of emotions proceeding downwards from *dark, sad, melancholy* at the top right corner, through *calm, soothing, dreamy*, to *sexy, fun, happy*, round to *feelgood, uplifting* and finally *angry, aggressive* towards the bottom left as reminiscent of Hevner's original arrangement (Figure 6.1). On the other hand, the location of clusters such as *delicate, intimate, peaceful, wistful* and *powerful, intense, moving, passionate, sensual, soulful* towards the centre of the map suggest that, while semantic organisation is strong in this space, it is not based on simple arousal-valence axes.

Table 6.2: Emotion words mapped onto a SOM

energetic funny quirky sunny		cute		angry feelgood power uplifting		driving		fun sexy	happy
cheerful fast positive		crazy			trance				chill
	joy		light weird				sweet		mellow
aggressive bright exciting			pure warm		nostalgic playful trippy				
dirty fiery joyous lush rousing								downtempo	relaxing
hypnotic majestic merry soaring triumphant witty wry	sweetness		passionate relaxed	meditative	mystical	gentle	ethereal	dreamy	
chilling emotive serene serious tranquil	earnest	delicate downbeat feeling sensual	peaceful yearning	wistful			atmospheric reflective	soothing	calm
inspiring		intimate		sleepy		sentimental			romantic
bitter noir poignant		soulful			intense	bittersweet			melancholic
comfort contemplative creepy crying depressed longing mournful	angst depressive gloomy sadness spiritual		moving powerful	deep	depressing haunting	emotional moody		dark	melancholy sad

Table 6.3: Emotion words mapped onto a SOM

depressive noir	creepy		emotive	meditative		mystical	gentle	reflective			depressing emotional moody			dark sad
gloomy	depressed			spiritual		playful			haunting					melancholy
mournful														melancholic
hypnotic soaring		longing	bitter	angst	feeling	powerful	intense		deep			atmospheric		calm
majestic	inspiring	chilling		sadness		moving			downbeat					soothing
serene tranquil triumphant		serious		crying			passionate sensual soulful				trance			dreamy ethereal
fiery merry				comfort contemplative poignant				warm						
rousing		lush	earnest		yearning			relaxed	sleepy					downtempo
joyous witty	joy		wry		delicate intimate	peaceful wistful					bittersweet sentimental			
exciting			bright					light	nostalgic			romantic		relaxing
dirty		sweetness			trippy									
cheerful positive			pure			weird	crazy	sunny			sweet		mellow	chill
funny								quirky		cute				
energetic														
fast	aggressive		angry		power		feelgood uplifting	driving			happy	fun		sexy

Multi-Dimensional Scaling

A more traditional visualisation technique is to project the emotion words into a very low-dimensional space by applying classical Multi-Dimensional Scaling (MDS) to the distance matrix $\mathbf{D_E}$ [Torgerson, 1958]. This approach highlights the relationship between the work presented in this and the preceding Chapters: where previously we attempted to learn a general latent semantic space from our matrix of track-word associations, here we attempt to model a subspace of emotion. In classical MDS, Principal Component Analysis is used to compute a low-rank approximation to the *doubly-centred* matrix of squared distances given by

$$\mathbf{B} = -\frac{1}{2}\mathbf{J}\mathbf{D}_{\mathbf{E}}^{(\mathbf{2})}\mathbf{J} \tag{6.2}$$

where $\mathbf{J} = \mathbf{I} - M^{-1}\mathbf{1}\mathbf{1}^T$ and M is the total number of emotion words. The coordinates of emotion words in a k-dimensional space are then given by $\mathbf{E_k}\mathbf{\Lambda_k}^{\frac{1}{2}}$, where $\mathbf{\Lambda_k}$ contains the k largest eigenvalues of $\mathbf{B}$, and $\mathbf{E_k}$ their corresponding eigenvectors.

MDS plots based on cosine distances between track vectors after LSA at rank 40, i.e. the same vectors used to train the SOM illustrated in Table 6.2, are given in Figures 6.3 and 6.4. The plots show the position of the emotion words in the first three dimensions found by MDS. The proportion of total variance explained by each dimension of the MDS solution is shown in Figure 6.5, suggesting that these are indeed the significant dimensions. Note that the proportion of variance accounted for by these dimensions is small in relation to comparable analyses in the literature: this is at least partly due to the very high dimensionality of our underlying data relative to the small samples acquired in a more conventional experimental setting.

A circumplex arrangement is strongly evident in Figure 6.4, with the x-axis (dimension 1 of the MDS solution) representing arousal, increasing from right to left, and the y-axis (dimension 3 of the MDS) representing valence. We also see the expected bipolar semantic pairings around the periphery of the mapping, with the positions of *peaceful, relaxed* opposite *fast, fiery*, and *depressive, dark* opposite *happy, cheerful* echoing the familiar quadrants of the circumplex. A feature of this mapping which differs from the traditional circumplex, however, is the empty region at the centre bottom of the plot. This can reasonably be interpreted as saying that neutral arousal is rarely associated with low valence in this emotional space, i.e. tag data suggests that negative feelings tend to be evoked by music which is also either exciting or relaxing, but not by music which is only moderately energetic. While this is hardly a controversial conclusion, it is noteworthy that it emerges so clearly from our data in comparison with previous work.

Figure 6.3, which shows dimension 2 of the MDS as the y-axis, shows another striking feature of the tag emotion space for music: there is a third significant dimension besides arousal and valence. The broadly triangular arrangement of words in this plane suggests that this third emotional dimension is particularly important for music associated with low arousal. Looking at the progression of words along the right-hand edge of the mappping, we see clearly that this dimension relates to the *spiritual, meditative* component of musical experience. The triangular shape of the mapping seems natural given this interpretation of dimension 2: a spiritual component is associated far more often with music that is slow and tranquil than with music that is fast and exciting. While the SOMs of Tables 6.2 and 6.3 have only two dimensions, they also show a clustering of words which supports the existence of a clear distinction between *sad* and *medidative* components in music associated with low arousal.

The space of musical emotion defined by social tags can consequently be described as having three significant axes, associated with the traditional bipolar scales of arousal and valence, as well as a third scale which we can designate as *transcendence*. While this echoes the three-dimensional arousal-valence-potence space reported in studies such as [Osgood *et al.*, 1957; Morgan & Heise, 1988], it is clearly different. Potence is described as a unipolar scale which distinguishes emotions with negative valence according to their association with feelings of power or powerlessness: thus *anger* and *fear* are strongly differentiated by their degree of potence. Transcendence, on the other hand, differentiates musical emotions with low arousal according to their degree of association with spiritual tranquility, ranging from *depression* (spiritual tension i.e. low transcendence) through simple *relaxation* to *meditation* (spiritual harmony i.e. high transcendence). The existence of a dimension of transcendence fills an important gap in the account of musical experience suggested by traditional two- and three-dimensional emotion spaces: the fact that listening to slow, sad music so often makes us feel good. Whether or not one accepts the details of interpretation offered here, the ability to model this particular emotional experience - so different from that, say, of looking at a series of sad faces - points to the value of the high-volume social media studied in this Thesis in uncovering significant aspects of music listening.

6.3 Emotion words and genre

Besides studying emotional responses to music in general, we can also use social tags to shed light on relationships between groups of pieces and specific words. In particular, we can measure the extent to which musical genres are associated with specific emotional vocabularies in the minds of listeners, and whether descriptions of emotion associated with particular genres go beyond clichés such as *aggressive* metal or *relaxing* classical music. While the experiments described in this section are simple, and based on a smaller dataset than we would want in order to draw robust conclusions, they are intended primarily to show how a statistical study of tags and other high-volume social media can be used in place of traditional questionnaire approaches to reach novel conclusions informing not only the cognitive or social psychology of music, but also musicological studies of so-called *music reception*, particularly in the domain of popular music.

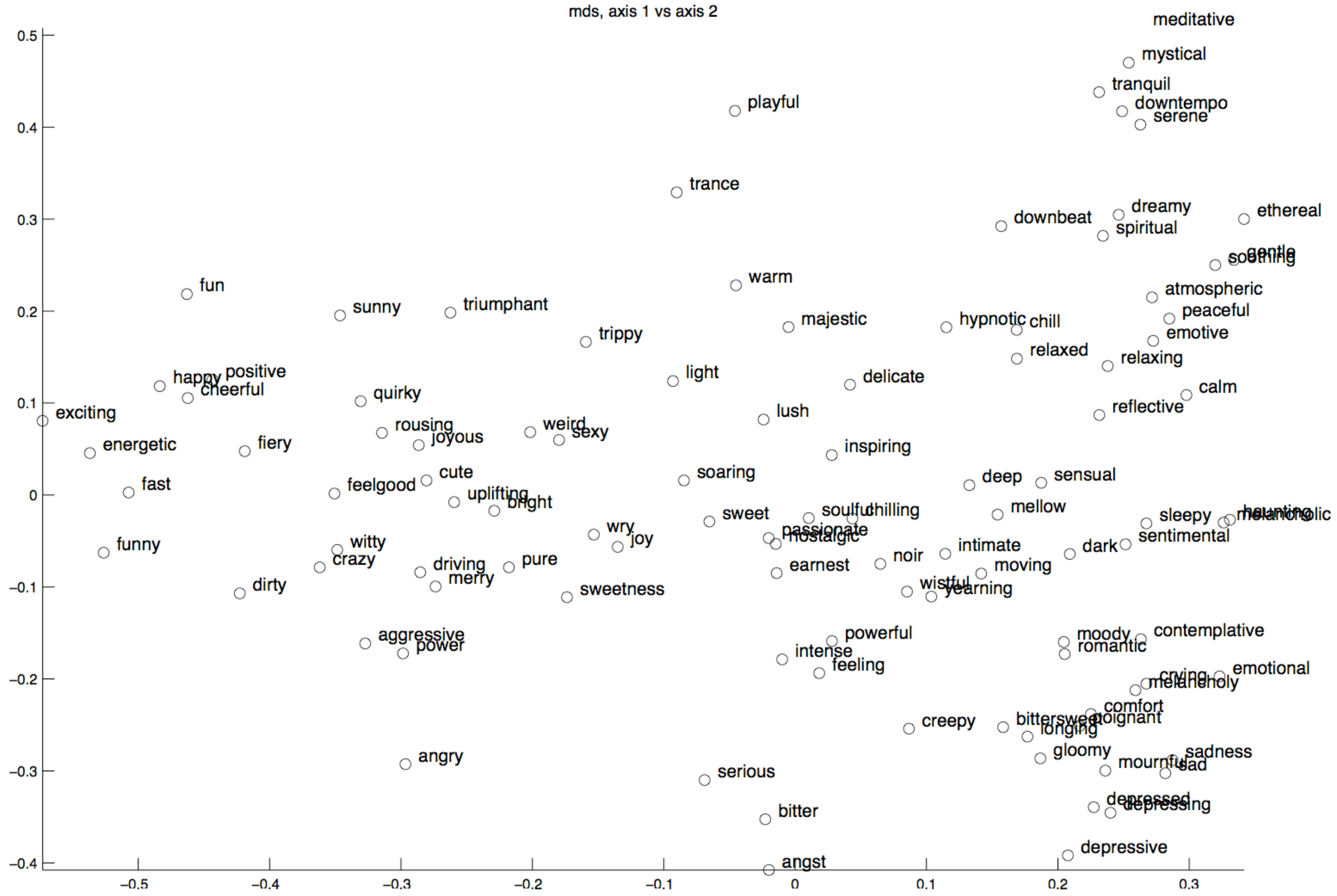


Figure 6.3: MDS of emotion words, dimensions 1 and 2

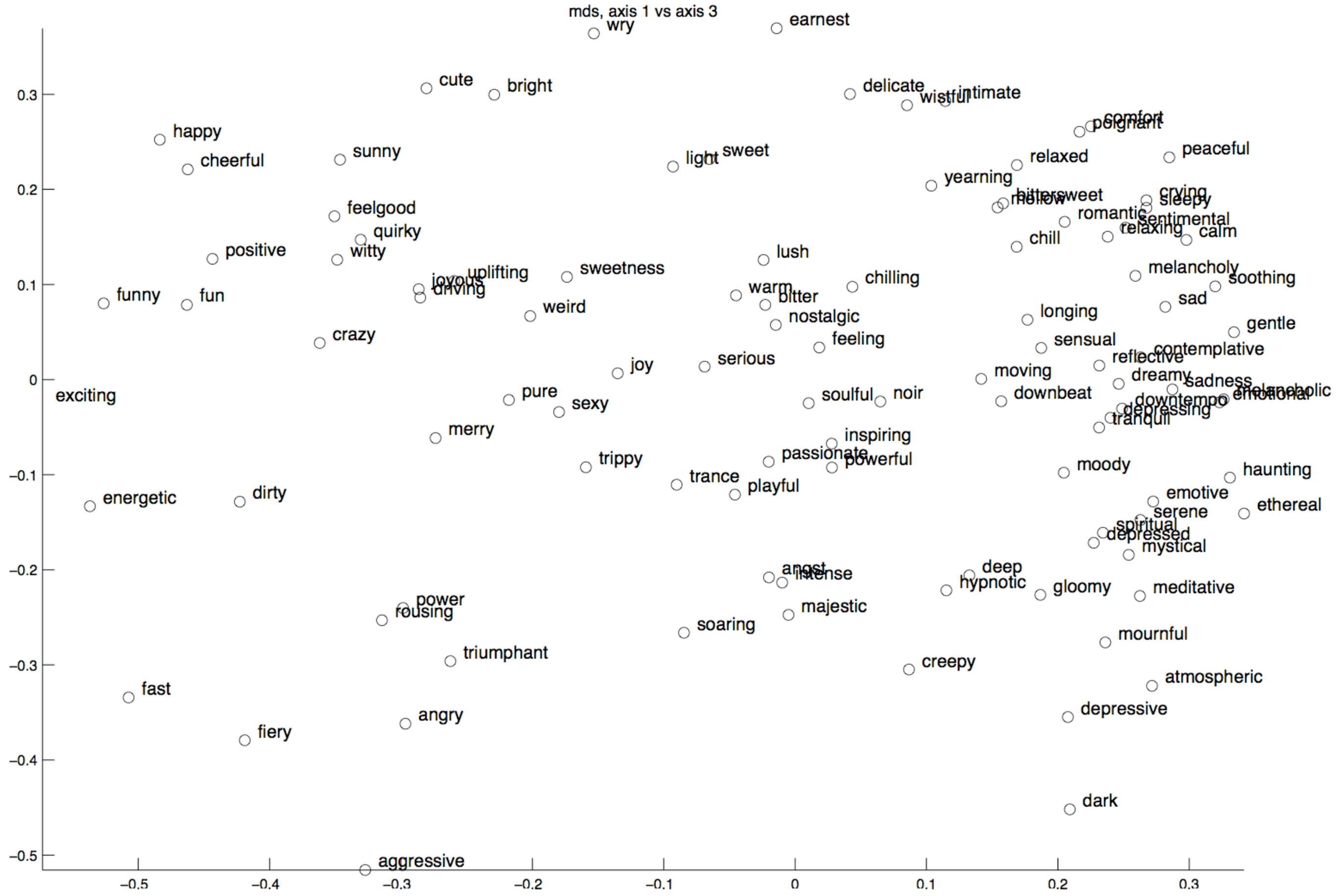


Figure 6.4: MDS of emotion words, dimensions 1 and 3

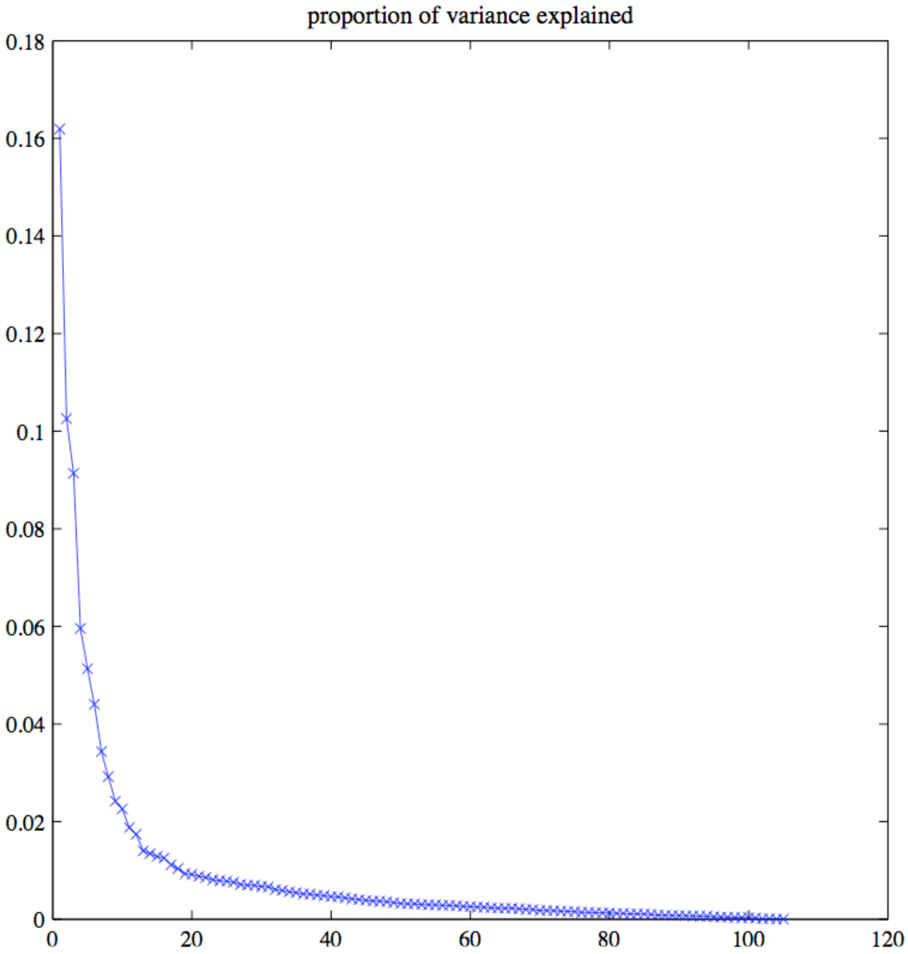


Figure 6.5: MDS of emotion words, variance explained

6.3.1 Do emotion words characterize genre?

We can test the extent to which genres are characterized by the use of specific emotion words with a simple information retrieval task similar to the ones used for evaluation in earlier chapters. Specifically we collect a set of tracks labelled by genre, and represent each track by the tag words applied to it, using a vector space model restricted to words in our emotion vocabulary. For each track in the dataset, we retrieve the nearest neighbouring tracks according to the model. We then measure the retrieval precision, i.e. the proportion of the retrieved tracks that were in the same genre as the query track. The precision will be high if tracks within the same genre are annotated with similar words. By restricting the vector space model to words in our emotion vocabulary only, we can use retrieval precision to measure the extent to which listeners characterize genres by their choice of emotion words. As a baseline we can measure precision with the vector space restricted to a randomly chosen set of words of the same size as our emotion vocabulary. If retrieval performance using the emotion vocabulary exceeds the baseline significantly, we can conclude that emotion annotations do indeed characterize artists and genres. By comparing with the retrieval performance using the entire vocabulary, we can get a measure of the extent to which genres are characterized by emotion words rather than other terms used in tagging.

Retrieval was evaluated over a subset of 1196 tagged tracks from the test set $\mathbf{T}$ described in Section 2.7, including between 4 and 12 tracks by each of 223 of the 224 artists assigned genres in [Knees, 2004]. Figure 6.6 shows precision-recall curves averaged over all the tracks for retrieval using, respectively, the emotion words only, all 11,509 words applied to these tracks, and an average over ten randomly chosen vocabularies of the same size as the emotion vocabulary. We observe that using the emotion words we retrieve twice the number of matching tracks as with a randomly chosen vocabulary, at all but the highest levels of recall. We observe further that the number of tracks of the same genre retrieved using the emotion words is around one third of the number retrieved

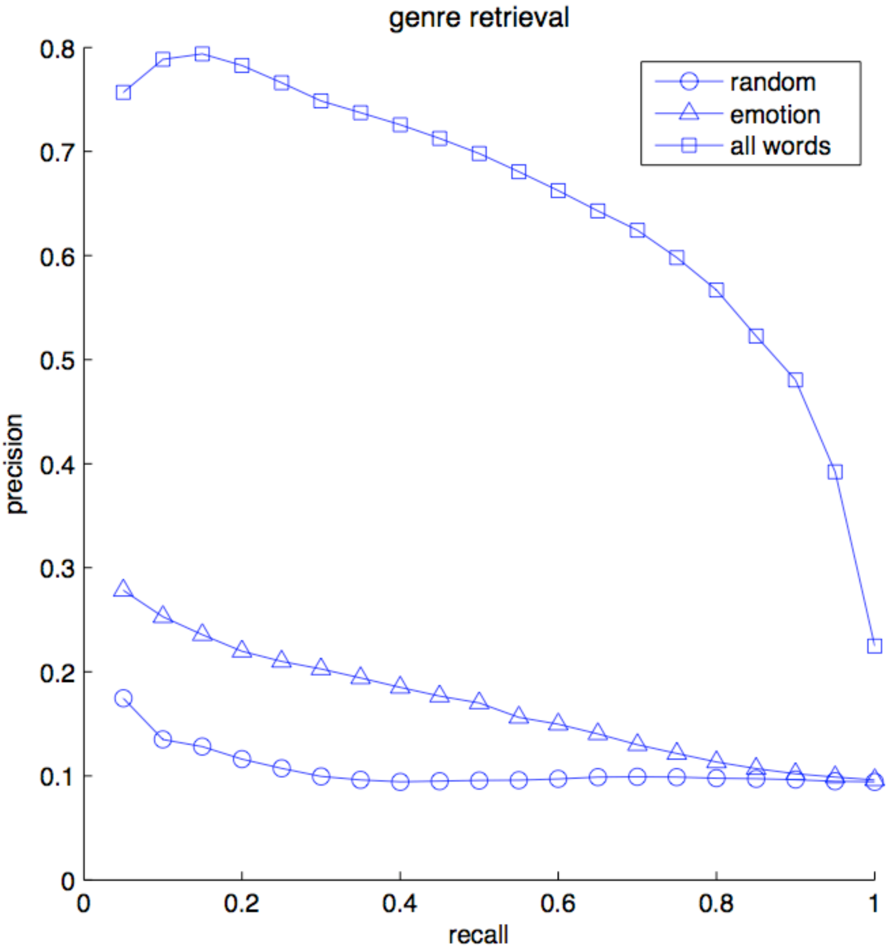


Figure 6.6: Genre retrieval using emotion words

using the full vocabulary, despite the fact that the emotion words make up less than 1% of the total, and that the full vocabulary includes words with a high information content for this task, such as artist names and genre labels themselves. While the retrieval task is clearly artificial in nature, the results show clearly that emotion words are far more powerful predictors of genre than randomly chosen vocabularies of equivalent size.

6.3.2 Which words are characteristic?

The retrieval results show that emotion words do characterize genre to some extent. We can inspect the particular emotion words that are characteristic by generating a simple emotion profile for each genre. We first make naive unsmoothed estimates of the genre-conditional

$$p(w|g) = \frac{n(w,g)}{\sum_{g'} n(w,g')} \tag{6.3}$$

and prior probabilities of each word w

$$p(w) = \frac{n(w)}{\sum_{w'} n(w')} \tag{6.4}$$

where $n(w,g)$ is the number of tags attached to tracks in genre g which contain w, and $n(w)$ the number of tags overall containing w. We then order the emotion words for each genre by their posterior probability

$$p(g|w) = \frac{p(w|g)p(g)}{p(w)} \tag{6.5}$$

i.e. the likelihood that genre of a track is g if we know that it has been tagged with word w. Finally we can estimate the overall predictability of each genre given all the emotion words applied to it from the contional entropy:

$$H(W_E|g) = \sum_{w \in W_E} p(w|g) log(\frac{1}{p(w|g)}) \tag{6.6}$$

Table 6.4 shows twelve top emotion words and their corresponding posteriors for each genre, as well as the conditional entropy.

The small set of tracks used here makes it unwise to draw sweeping conclusions from the profiles in Table 6.4, because the significance of individual words can easily be overestimated within a small dataset. It is nonetheless interesting that, within the tagging community at least, indie and folk tracks appear to be particularly characterised by the emotion words associated with them

Table 6.4: Top emotion words by genre

genre $H(W\|g)$	word $p(g\|w)$
alt/indie (40.8)	soaring (0.71) creepy (0.67) majestic (0.60) melancholic (0.54) depressing (0.53) moody (0.53) serious (0.50) dark (0.46) angst (0.44) melancholy (0.39) depressive (0.39) haunting (0.37)
folk (31.8)	poignant (0.73) mystical (0.67) wry (0.60) intimate (0.57) yearning (0.52) comfort (0.50) joyous (0.50) noir (0.50) wistful (0.50) spiritual (0.40) bitter (0.38) bittersweet (0.36)
electronic (29.5)	trance (0.82) downbeat (0.57) downtempo (0.55) hypnotic (0.50) meditative (0.50) tranquil (0.50) soothing (0.48) chilling (0.46) peaceful (0.44) warm (0.43) delicate (0.43) joy (0.43)
pop (25.7)	lush (0.45) emotive (0.44) cute (0.38) inspiring (0.36) sexy (0.35) longing (0.29) positive (0.26) bright (0.25) fun (0.24) energetic (0.24) cheerful (0.23) playful (0.23)
rnb/soul (23.4)	rousing (0.57) soulful (0.54) earnest (0.50) fiery (0.50) passionate (0.30) sensual (0.28) hypnotic (0.25) yearning (0.24) downbeat (0.21) inspiring (0.21) mournful (0.20) triumphant (0.20)
heavy metal (15.9)	aggressive (0.57) fiery (0.50) power (0.45) angry (0.28) depressive (0.28) fast (0.25) deep (0.24) feelgood (0.18) pure (0.17) uplifting (0.16) intense (0.15) driving (0.14)
jazz (15.7)	serene (0.33) delicate (0.29) gloomy (0.25) tranquil (0.25) majestic (0.20) triumphant (0.20) sentimental (0.16) longing (0.14) rousing (0.14) wistful (0.13) spiritual (0.12) gentle (0.11)
punk (14.2)	noir (0.25) exciting (0.23) wry (0.20) pure (0.19) angst (0.19) comfort (0.17) fast (0.17) feeling (0.16) funny (0.14) playful (0.14) bittersweet (0.13) depressed (0.11)
rap/hiphop (12.2)	mournful (0.40) dirty (0.23) triumphant (0.20) witty (0.20) crazy (0.19) serious (0.19) creepy (0.17) bitter (0.15) funny (0.14) angry (0.12) driving (0.11) cheerful (0.09)
country (11.0)	longing (0.29) earnest (0.25) reflective (0.24) gloomy (0.21) relaxed (0.19) passionate (0.18) crying (0.17) bittersweet (0.16) emotive (0.11) lush (0.09) spiritual (0.08) angst (0.07)
rock (10.1)	crying (0.50) exciting (0.23) quirky (0.20) rousing (0.14) fast (0.10) uplifting (0.10) witty (0.10) positive (0.09) light (0.08) fun (0.08) happy (0.07) energetic (0.07)
classical (8.3)	meditative (0.50) serene (0.33) bright (0.25) romantic (0.20) joyous (0.17) sensual (0.12) contemplative (0.11) emotive (0.11) gentle (0.11) peaceful (0.11) mournful (0.10) powerful (0.10)
reggae (5.8)	chilling (0.23) serious (0.19) emotive (0.11) dirty (0.09) sleepy (0.07) chill (0.06) gentle (0.06) feeling (0.05) nostalgic (0.05) cheerful (0.05) playful (0.05) cute (0.04)
blues (2.9)	mystical (0.33) sensual (0.08) bitter (0.08) sadness (0.07) dirty (0.05) feeling (0.03) sentimental (0.02) soulful (0.02) passionate (0.02) chill (0.02) nostalgic (0.02) intense (0.01)

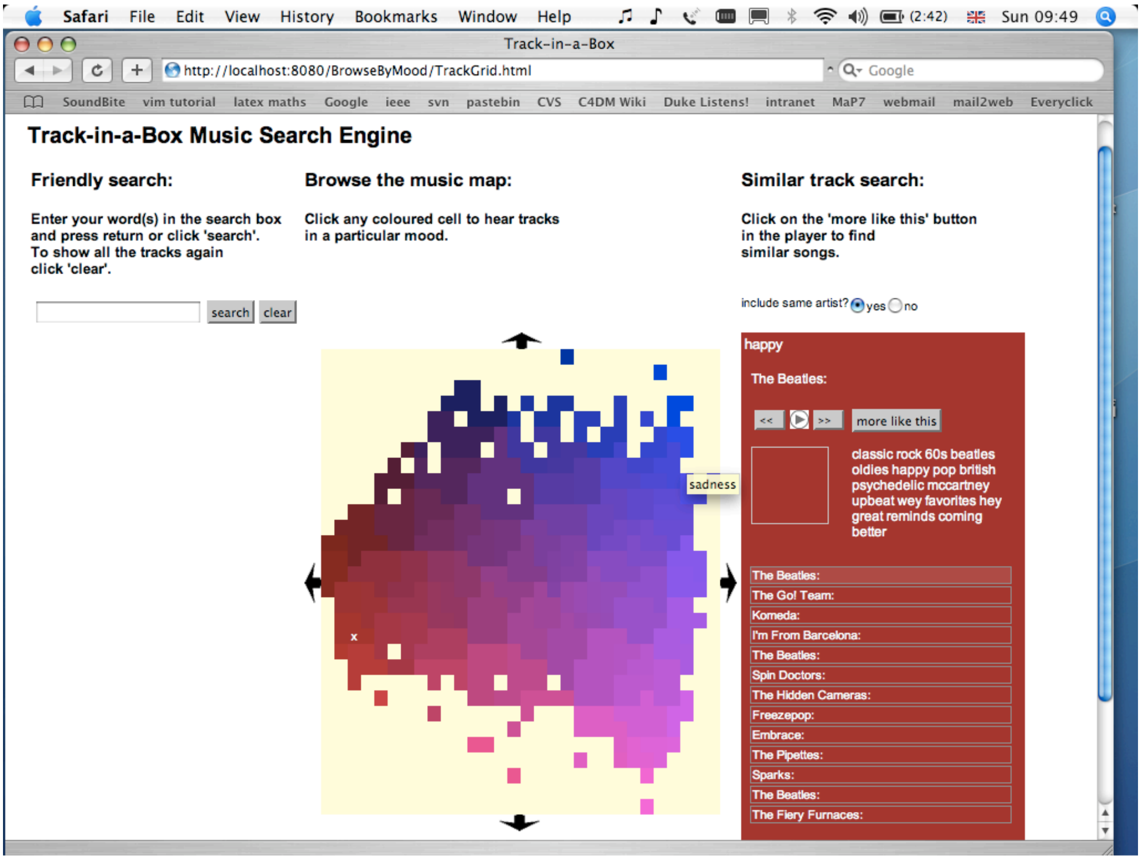
in comparison to other genres. Metal is indeed characteristically described as *aggressive*, even though we can reasonably assume that a large proportion of tags applied to metal tracks come from metal-lovers. Classical music is distinguished to an extent as *meditative, serene, contemplative*, but also as *bright, romantic, joyous, sensual*, suggesting that the tagging community includes listeners keen to give sophisticated responses to classical music.

Note that while these observations are offered with caution, in a dedicated study it would be straightforward to support robust conclusions of this kind by using a larger dataset and in particular by cross-validation between distinct sets of artists or tracks.

6.4 Browsing the semantic space of musical emotion

The results of Section 6.2 demonstrate that straightforward computational methods applied to social tags can be used to map large numbers of tracks into psychologically meaningful two- and three-dimensional spaces. Besides being of theoretical interest, such mappings also offer a valuable paradigm for interfaces to large music collections, to serve users looking to browse large numbers of tracks by mood. Such interfaces could be valuable both to music lovers looking to find background music for different social occasions or activities, and to users with a commercial aim in mind, such as film or television producers looking for music to match particular dramatic or visual scenario, or to associate a particular mood with some product to be advertised.

Figure 6.7 shows a screenshot of Trackinabox, a system developed using the work reported in this Thesis. Trackinabox incorporates a map view of a collection of tracks, using an MDS mapping similar to those of Section 6.2. The collection can also be searched by keyword, using the methods of Chapter 5, and similar tracks to any of those found by mood or by keyword can then be retrieved as in the experiments reported in Chapter 2. Trackinabox has not


Safari File Edit View History Bookmarks Window Help
(2:42) Sun 09:49
Track-in-a-Box
http://localhost:8080/BrowseByMood/TrackGrid.html
Google
SoundBite vim tutorial latex maths Google ieee svn pastebin CVS C4DM Wiki Duke Listens! intranet MaP7 webmail mail2web Everyclick
search
clear
sadness
x
<<
>>
more like this

Track-in-a-Box Music Search Engine

Friendly search:

Enter your word(s) in the search box
and press return or click 'search'.
To show all the tracks again
click 'clear'.

Browse the music map:

Click any coloured cell to hear tracks
in a particular mood.

Similar track search:

Click on the 'more like this' button
in the player to find
similar songs.

include same artist? ● yes ○ no

happy

The Beatles:

classic rock 60s beatles oldies happy pop british psychedelic mccartney upbeat wey favorites hey great reminds coming better

The Beatles:
The Go! Team:
Komeda:
I'm From Barcelona:
The Beatles:
Spin Doctors:
The Hidden Cameras:
Freezepop:
Embrace:
The Pipettes:
Sparks:
The Beatles:
The Fiery Furnaces:

Figure 6.7: Track-in-a-box

been formally evaluated, and is consequently not reported in more detail here.

The mappings of Section 6.2 are also used as the basis of a novel interactive spatial audio interface that positions tracks in a virtual 3-d space around the listener, to be navigated with the use of a small games controller: this interface, together with the results of a small user study, is reported in detail in [Stewart *et al.*, 2008].

The results of the user study, as well as experience gained during the development of the Trackinabox software, suggest the potential usefulness of being able to visualise emotion words and individual tracks within the same space. The following subsections therefore investigate the application to tags of a further visualisation technique, Correspondence Analysis, which does indeed allow us to map words and tracks into a single low-dimensional space.

6.4.1 Correspondence Analysis

Correspondence Analysis (CA) is a well-established technique of dimension reduction used primarily for visualising multivariate categorical data [Benzécri, 1977; Greenacre, 1984]. It has two properties that make it extremely attractive for our purposes:

1. it enables the visualisation of two sets of cross-tabulated variables (in our case tracks and semantic terms) in the same low-dimensional space;
2. Euclidean distances in the visualisation represent distributional (χ^2) distances in the data.

CA is a generalised form of Principal Component Analysis suitable for application to an M by N table of co-occurrence data $\mathbf{F}$, where $\mathbf{F}$ has been normalised to have total sum 1. CA finds a low-dimensional projection of $\mathbf{F}$ which optimally preserves χ^2-distances between row and column *profiles*

$$\begin{aligned} \mathbf{f}^{c|r=i} &= \left(\frac{f_{i1}}{f_i}, ..., \frac{f_{iN}}{f_i}\right) \\ \mathbf{f}^{r|c=j} &= \left(\frac{f_{1j}}{f_j}, ..., \frac{f_{Mj}}{f_j}\right) \end{aligned}$$

where f_i, f_j are the row and column sums respectively, i.e. $f_i = \sum_{j=1}^{N} f_{ij}$ and $f_j = \sum_{i=1}^{M} f_{ij}$.

The χ^2-metric between row profiles is a weighted Euclidean distance where the weight for each column is given by $\frac{1}{f_j}$; the metric between column profiles is weighted similarly by $\frac{1}{f_i}$. The χ^2-metric has the desirable property that distances between columns (tag words) do not change if columns (tracks) with identical profiles (normalised term vectors) are amalgamated, and vice versa.

We compute a generalised SVD of $\mathbf{F}$

$$\tilde{\mathbf{F}} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}' \tag{6.7}$$

where $\mathbf{\Delta}$ is a diagonal matrix, and $\mathbf{U}$ and $\mathbf{V}$ satisfy

$$\mathbf{U}'(\mathbf{F^r})^{\mathbf{-1}}\mathbf{U} = \mathbf{V}'(\mathbf{F^c})^{\mathbf{-1}}\mathbf{V} = \mathbf{I} \tag{6.8}$$

where $\mathbf{F_r}$ and $\mathbf{F_c}$ are diagonal matrices of the row and column sums respectively. Co-ordinates $\mathbf{S}$ of row profiles onto axes $\mathbf{U}$ are then given by

$$\mathbf{f}^{c|r} = \mathbf{US} \tag{6.9}$$

where

$$\mathbf{S} = \mathbf{\Delta}\mathbf{V}'(\mathbf{F^c})^{\mathbf{-1}} \tag{6.10}$$

Co-ordinates $\mathbf{T}$ of column profiles onto axes $\mathbf{V}$ are given similarly by

$$\mathbf{f}^{r|c} = \mathbf{VT} \tag{6.11}$$

where

$$\mathbf{T} = \mathbf{\Delta}\mathbf{U}'(\mathbf{F^r})^{\mathbf{-1}} \tag{6.12}$$

Row and column profiles can then be plotted in the same d-dimensional space, taking only the first d co-ordinates of $\mathbf{S}$ and $\mathbf{T}$. Although it is not meaningful in general to interpret row-column distances in this visualisation, it does show the relative distances of a single row (track) to all the columns (emotion words), and vice versa.

This suggests a natural application of CA with $d = 2$ to create a browse-by-mood interface to a collection of tracks, using a normalised portion of the document-term matrix, with row profiles representing tracks and columns restricted to mood terms. The resulting plot of tracks and terms shows mood words in a meaningful relationship, while tracks in any particular region of the space should be well described by nearby words.

6.4.2 Evaluation

While a full user evaluation of the application of CA remains for future work, this approach was tested empirically on a small list of 14 mood words, consisting of the subset of terms from Hevner's original list of musical emotions [Hevner, 1936] which were applied to at least 50 tracks in the dataset, and the subset of 3176 tracks tagged with at least one of these words. Figure 6.8 shows the resulting positions of the terms and tracks. The organisation of the plot can be evaluated by calculating the mean AP for each mood word, where we consider a track to be relevant to its closest mood word in the plot if it has been tagged with it.

To comply with the allowable interpretation of distances in CA, we take the mean AP for each term only over tracks which are closer to it in the CA space than they are to any other term (so each track in the dataset gets considered exactly once). The results are given in Table 6.5, showing that the plot partitions the space almost perfectly by this measure. It is important to note that precision is measured here against words found in tags themselves, not a verifiable external source of information. Nonetheless these results suggest that CA is worthy of further investigation as the basis of practical browse-by-mood interfaces to music collections.

6.5 Conclusions

This Chapter presented studies of emotion words in tags from a variety of perspectives, based only on the assumption that it is reasonable to treat social tags as unconstrained verbal responses to tracks. While such responses are subject to frequent inconsistency, noise, and even deliberate spam, this Chapter suggested methods to extract valuable information from tags as a complement to more traditional laboratory experiments from the domain of music psychology, and presented a range initial results to give a flavour of the work possible with this novel approach. An updated vocabulary of emotion words for music

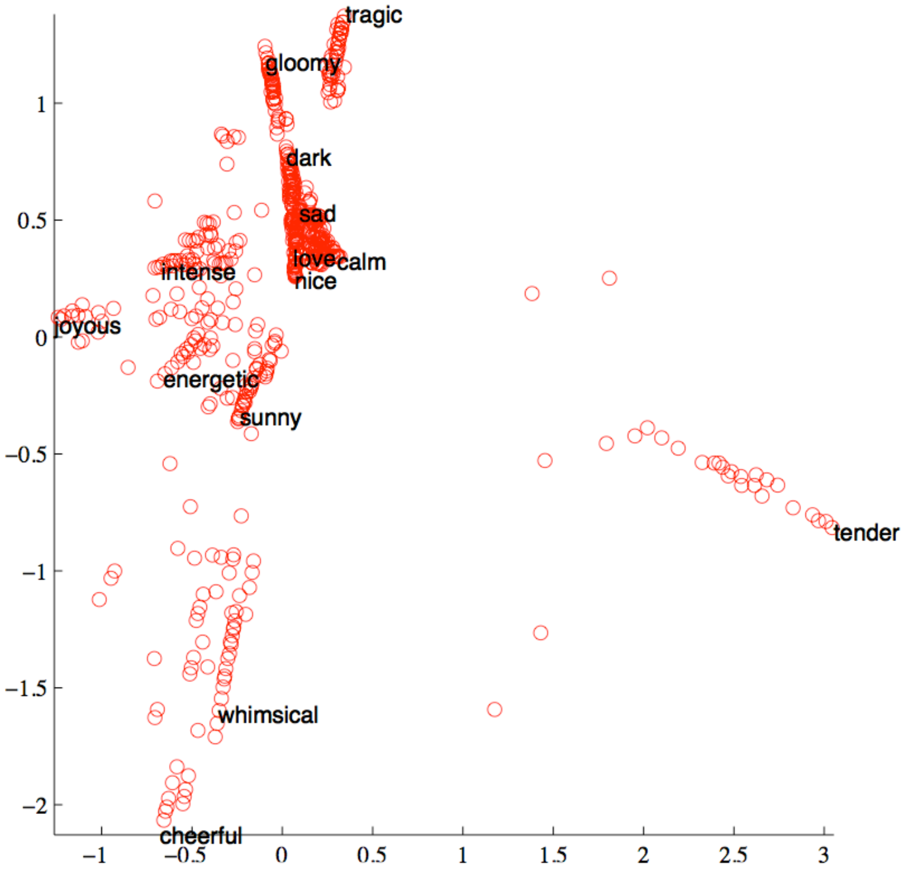


Figure 6.8: CA joint plot of mood words and tracks

Table 6.5: Mean Average Precision for mood words

Mood	**mean AP**
calm	0.998
cheerful	1.000
dark	0.947
energetic	0.925
gloomy	0.987
intense	0.924
joyous	1.000
love	1.000
nice	0.939
sad	0.965
sunny	0.942
tender	1.000
tragic	1.000
whimsical	0.919

was first mined from tags, with the help of some expert selection, and shown to be significantly different from that found in recent laboratory work. Emotion words from tags were mapped into a contemporary Circumplex 2.0 using two different visualisation techniques, suggesting the existence of a possible additional dimension of *transcendence* in the emotion space for music in addition to the well-known axes of arousal and valence. The association between emotion words and musical genres was investigated and quantified, showing that emotion words in general are at least weakly predictive of genre, and tag data was used to generate characteristic emotion profiles for several well-known musical genres. Finally Correspondence Analysis applied to word occurrences in tags was proposed as a novel basis for a user interface to large collections of tracks, allowing users to browse easily in a psychologically-motivated space structured jointly around emotion words and tracks themselves.

While these studies are all preliminary in nature, and based on relatively small datasets, they demonstrate the scope of possible future research into emotion based on a study of social tags for music, and indeed the possible applications of scientific studies of social media in general to social and cognitive psychology, as well as to musicology and its sister disciplines in the humanities.

Chapter 7

Conclusions

Over and above any of the individual results reported here, the work reported in this thesis has aimed to encourage a modest but significant paradigm shift in the study of music information retrieval. In place of the audio classification tasks that have dominated research in the field for many years, this thesis proposes the use of frameworks and models familiar from conventional text IR as more appropriate and effective for real-world applications. Such a shift is desirable - and indeed possible in the first place - due to the massive increase in the availability of concise descriptive information associated with specific audio tracks resulting from the very recent phenomenon of social tagging. The wide and growing availability of tags clearly favours approaches that combine audio and descriptive information, rather than attempting simply to infer one from the other. The nature of tags for music, indeed of any kind of writing about music, also strongly suggests that to best support useful retrieval tasks we should learn significant semantic aspects from our data rather than attempting to model predetermined categories such as genre labels.

7.1 Summary

The key findings reported in this thesis are summarised briefly in the following sections.

7.1.1 The semantics of social tags for music

A dataset of over half a million social tags was collected and analysed, revealing that tags for music resemble natural language rather than a rigid lexicon of class labels: human descriptions of music are informal, discursive and personal, and the vocabulary of tags grows in accordance with Heaps' Law. Chapter 2 describes how nonetheless semantic analysis of tags using the methods of classical Information Retrieval, such as Latent Semantic Analysis and aspect modelling, reveals coherent latent structure embodying concepts such as genre, era, mood, instrumentation and nationality . Simple IR methods applied to tags were also shown to outperform the use of both audio features and web-mined text by a huge margin on standard retrieval tasks.

7.1.2 A discrete representation for musical audio

A discrete representation was developed in Chapter 3 to simplify the use of IR methods to model audio content jointly with social tags. This takes the form of a vocabulary of *muswords* representing musically significant regions of recordings of, in principle, the entire universe of music. The regions themselves are automatically identified using a novel method which detects the start of new sections within a track. In contrast to established methods for structural segmentation, this process can be accomplished in a single pass, making it scalable to large collections of audio. A representation using muswords alone was shown to outperform alternative approaches to discretizing audio content when used as a basis for standard retrieval tasks.

7.1.3 Semantic models for music annotation and retrieval

Semantic models based on a joint vocabulary of words and muswords were developed in Chapter 4 and applied to retrieval tasks in a realistic setting described in Chapter 5, in which some tracks were only sparsely tagged and others were completely untagged. Retrieval performance was shown to be significantly improved both by the use of audio information in addition to tags, and specifically by modelling latent structure in the semantic space. A simple method was introduced to use the same model to generate annotations for untagged tracks. Although evaluation of open vocabulary annotation is difficult, results were presented suggesting that these annotations are at least as accurate as those produced by the closest comparable system.

7.1.4 Emotion words in social tags for music

The semantic models used previously for practical tasks were employed in Chapter 6 to uncover latent patterns in the application of mood words in social tags. In particular two different visualization methods were employed to embed mood words from tags in very low-dimensional spaces, revealing structures corresponding strikingly to the well-established dimensional theory of affect, though with some interesting differences to traditonal models. A joint embedding of mood words and tracks based on Correspondence Analysis was proposed as the basis for a novel music browsing interface, with encouraging initial evaluation results.

7.2 Future work

Many aspects of the research reported in this thesis could be improved or extended: the following sections attempt to identify the most promising opportunities for further work.

7.2.1 Data collection

Data collection remains the single most important step in building semantic systems for music retrieval, and two straightforward measures should be taken as a basis for future research. Firstly the easy availability of distributed storage and processing infrastructure now makes it possible to work with datasets that are several orders of magnitude larger than those used in the experiments reported in Chapters 2 to 6. Secondly, as outlined in Section 5.4, some simple cleaning procedure should be applied to remove the most obviously noisy tags.

With hindsight the very simple model of Equation 2.2 used to interpret the publicly available 'counts' for tag assignments as semantic weights was probably too naive, and more sophisticated approaches should be considered, for example by taking into account the overall frequency of the tag in question, as well as information about the reach of the track or artist to which it is applied. It would also make sense to add artist names explicitly to the tags attached to each track, as these are already commonly used by listeners in practice: this should add semantic richness to learned models, for example allowing us to label latent aspects with the names of artists, as well as allowing new avenues for evaluation. Finally, assuming a sufficiently large experimental dataset, it would be sensible to reconsider the decision made in Section 2.2 to model individual words, and consider modelling either entire tags, or at least recurring ngrams of words.

7.2.2 Modelling

There are many ways in which modelling of audio content could be improved, even while keeping to the existing framework where we construct a sparse representation based on a vocabulary of discrete muswords extracted from specific regions of interest in each track.

The various stages making up the method of Section 3.1 used to find the regions of interest should be more fully investigated and evaluted. In particular a wider range of low-level audio features should be considered in place of MFCCs and simple rhythmic features, and statistics other than mean and variance should be investigated when summarising them. One immediate possibility is to take advantage of recent work on audio similarity, and replace simple low-level feature statistics with GMM supervectors [Charbuillet *et al.*, 2011], which capture timbral characteristics of individual tracks relative to a Universal Background Model trained on a large corpus of audio: this is reported to improve both performance and scalability, as the resulting song features are comparable with Euclidean distance.

More sophisticated methods should also be used to establish significant novel section boundaries, for example by combining two separate measures, one expressing local contrast and the other overall novelty within a song. The effect of varying the number of distinct muswords in the overall vocabulary should also be fully investigated. While evaluating the audio representation on retrieval tasks remains sensible given an ultimate goal of supporting search systems, it would also be interesting to evaluate separately the part of the process concerned with finding regions of interest, for example by formalising the listening test of Section 3.2.1 to ensure that the extracted regions do indeed sound similar. If successful, this approach would also have direct application to audio thumbnailing for collection browsing.

The aspect model of Chapters 3 to 5 could be improved as suggested in [Hofmann, 2001], for example by using a training algorithm in which the learning rate is *tempered* in place of early stopping; by training several distinct models with different numbers of aspects and interploating their results at query time; or by applying *pseudo-tfidf* reweighting to promote the significance of rare aspects at query time. On the other hand it is also tempting simply to replace the aspect model altogether with a topic model learned by Latent Dirichlet Allocation [Blei *et al.*, 2003], which is reported to give better performance on text IR tasks, and for which a highly scalable implementation is now freely available [Smola & Narayanamurthy, 2010].

7.2.3 Applications and evaluation

Given the reported weakness of current benchmark training sets and offline metrics for semantic annotation and retrieval [Law *et al.*, 2009; Marques *et al.*, 2011], perhaps the most significant opportunity for further work is in the development of applications allowing human evaluation of semantic models and corresponding algorithm performance. This could take the form of further development of the existing Trackinabox prototype mentioned briefly in Section 6.4, or the spatial audio interface described in [Stewart *et al.*, 2008]: the former could naturally be developed as a tablet app allowing browsing and playback.

Recent experience suggests, however, that a more efficient approach to gathering evaluation data is via a direct online questionnaire that also offers some simple form of engagement, for example by playing music that the user likes, and at least a nominal reward for participation, such as points on a leaderboard [Levy, 2011]; such an application can also be designed with little overhead to make use of well-established methods for controlled evaluation of rival algorithms [Kohavi *et al.*, 2007]. Semantic search of a music catalogue by free text query clearly lends itself to this approach, as the UI requirements are largely trivial, and would involve very limited development effort compared to existing online annotation games; on the other hand, if hosted in a suitable context it should avoid the issues of spam and low quality responses associated with soliciting online questionnaire answers for money [Lee, 2010; Speck *et al.*, 2011].

7.3 Reflections

Even during the lifetime of this thesis, there have been some encouraging signs that the focus of MIR research is maturing from simplistic classification paradigms to engage with richer and more realistic problems: for example in the 2011 round of the MIREX algorithm contest the simple Train-Test set of classification tasks was relegated to 'DIY' status, with the results no longer reported on the competition website or formal results poster[1]. On the other hand, semantic approaches to music retrieval and annotation have gained relatively little attention, the most significant contributions being [Law *et al.*, 2010], which builds on the methods proposed in [Levy & Sandler, 2007, 2009], and the neural networks described in [Mandel *et al.*, 2011b; Hamel *et al.*, 2011], which attempt to capture semantic relationships implicitly in learned features. Semantic search as a paradigm for music discovery still also has few real world implementations, despite rapid growth in streaming services backed by large catalogues, and continuing demand for potentially lucrative systems to match musical content to films, TV programmes or adverts [Inskip *et al.*, 2010].

Some developments within the past few months suggest that we may finally be starting to see the transition of these ideas to the mainstream. A paper reporting ongoing research at Google [Weston *et al.*, 2011] describes a novel learning framework in which audio features, tags, artist names, and in principle any other labels of interest, are mapped into a single joint semantic space: the learned mappings ensure that both similar sounding songs and related labels lie close together. Any number of labelling or retrieval tasks can then be performed based simply on the distance between relevant entities in this space. Besides the elegance and reported effectiveness of this approach, this work potentially has a significant impact beyond the research community as it was designed to support Google's own music streaming service[2].

The recent release of a research dataset of some 8.5 million tags by Last.fm

[1] http://www.music-ir.org/mirex/wiki/2011:MIREX2011_Results, http://www.music-ir.org/mirex/results/2011/mirex_2011_poster.pdf

[2] http://music.google.com, personal communication from Doug Eck, October 2011

[Bertin-Mahieux *et al.*, 2011] should make it possible for researchers with more limited resources to work on similarly scalable methods for semantic annotation and retrieval. Finally a new user interface to a catalogue of several million tracks by unsigned artists[3], developed for Last.fm by a team including the author of this thesis, should introduce a semantic search paradigm to a large existing community of music listeners, as well as enabling ongoing research based on day-to-day feedback from a working large-scale system.

[3] `http://www.last.fm/discover`

Bibliography

Aberer, K., Cudr'e-Mauroux, P., Ouksel, A. M., Catarci, T., Hacid, M.-S., Illarramendi, A., Kashyap, V., Mecella, M., Mena, E., Neuhold, E. J., Troyer, O. De, Risse, T., Scannapieco, M., Saltor, F., Santis, L. De, Spaccapietra, S., Staab, S., & Studer, R. 2004. Emergent Semantics Principles and Issues. *In: Proc. 9th International Conference on Database Systems for Advanced Applications (DASFAA 2004)*.

Aucouturier, J.-J. 2006. *Ten experiments on the modelling of polyphonic timbre*. Ph.D. thesis, University of Paris 6.

Aucouturier, J.-J., & Pachet, F. 2004. Improving timbre similarity: How high is the sky? *Journal of Negative Results in Speech and Audio Sciences*, **1**(1), 1–12.

Aucouturier, J.-J., Defreville, B., & Pachet, F. 2007. The bag-of-frame approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music. *Journal of the Acoustical Society of America*, **122**(2), 881–891.

Barnard, K., & Forsyth, D.A. 2001. Learning the Semantics of Words and Pictures. *In: Proc. IEEE International Conference on Computer Vision*.

Barrington, Luke, Chan, Antoni, Turnbull, Douglas, & Lanckriet, Gert. 2007. Audio Information Retrieval using Semantic Similarity. *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*.

Baumann, S., & Hummel, O. 2003. Using cultural metadata for artist recommendations. *In: Proc. 3rd International Conference on Web Delivering of Music (WEDELMUSIC 2003).*

Bello, J. P., Duxbury, C., Davies, M. E., & Sandler, M. B. 2004. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, **11**(6), 553–556.

Benzécri, J.-P. 1977. Histoire et préhistoire de l'analyse des données. *Cahiers de l'Analyse des Données*, **2**, 9–40.

Bergstra, James, Mandel, Michael I., & Eck, Douglas. 2010. Scalable genre and tag prediction with spectral covariance. *In: Proc. 11th International Society for Music Information Retrieval Conference.*

Bertin-Mahieux, T., Eck, D., & Mandel, M. 2010. Automatic Tagging of Audio: The State-of-the-Art. *In:* Wang, Wenwu (ed), *Machine Audition: Principles, Algorithms and Systems*. IGI Publishing.

Bertin-Mahieux, Thierry, Ellis, Daniel P.W., Whitman, Brian, & Lamere, Paul. 2011. The Million Song Dataset. *In: Proc. 12th International Society for Music Information Retrieval Conference.*

Bicknell, J. 2002. Can music convey semantic content? A Kantian approach. *Journal of Aesthetics and Art Criticism*, **60**(3), 253–261.

Blei, D., & Jordan, M. 2003. Modeling Annotated Data. *In: Proc. 26th Annual International ACM SIGIR Conference.*

Blei, David M., Ng, Andrew Y., & Jordan, Michael I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, **3**, 993–1022.

Cano, P., Gómez, E., Gouyon, F., Herrera, P., Koppenberger, M., Ong, B., Serra, X., Streich, S., & Wack, N. 2006. *ISMIR 2004 Audio Description Contest*. Tech. rept. Music Technology Group, Universitat Pompeu Fabra.

Carneiro, G., Chan, A.B., Moreno, P.J., & Vasconcelos, N. 2007. Supervised learning of semantic classes for image annotation and retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **29**(3), 394–410.

Chai, W., & Vercoe, B. 2003. Music Thumbnailing via Structural Analysis. *In: Proc. 11th ACM International Conference on Multimedia*.

Charbuillet, Christophe, Tardieu, Damien, & Peeters, Geoffroy. 2011. GMM supervector for Content Based Music Similarity. *In: Proceedings of the 14th International Conference on Digital Audio Effects*.

Collier, G. L. 2007. Beyond valence and activity in the emotional connotations of music. *Psychology of Music*, **35**(1), 110–131.

Davies, M., & Plumbley, M. 2008. Exploring the effect of rhythmic style classification on automatic tempo estimation. *In: Proc. 16th European Signal Processing Conference (EUSIPCO 2008)*.

Deerwester, S., Dumais, S., Furnas, G. W., Landauer, T. K., & Harshman, R. 1990. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, **41**(6), 391–407.

Desmet, P. M. A. 2008. Product Emotion. *Pages 379–397 of:* Schifferstein, H.N.J., & Hekkert, P. (eds), *Product experience*. Elsevier.

Downie, J. S., West, K., Ehmann, A., & Vincent, E. 2005. The 2005 Music Information Retrieval Evaluation Exchange (MIREX 2005): Preliminary overview. *In: Proc. 6th International Society for Music Information Retrieval Conference*.

Eck, Douglas, Lamere, Paul, Bertin-Mahieux, Thierry, & Green, Stephen. 2008. Automatic Generation of Social Tags for Music Recommendation. *Pages 385–392 of:* Platt, J.C., Koller, D., Singer, Y., & Roweis, S. (eds), *Advances in Neural Information Processing Systems 20*. Cambridge, MA: MIT Press.

Farnsworth, P. R. 1954. A study of the Hevner adjective list. *Journal of Aesthetics and Art Criticism*, **13**, 97–103.

Farnsworth, P. R. 1969. *The social psychology of music*. Iowa State University Press.

Fellbaum, C. (ed). 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Freund, Y., & Shapire, R.E. 1996. Experiments with a new boosting algorithm. *In: Proc. 13th International Conference on Machine Learning*.

Fu, Zhouyu, Lu, Guojun, Ting, K M, & Zhang, Dengsheng. 2011. A Survey of Audio-based Music Classification and Annotation. *IEEE Transactions on Multimedia*, **13**(99), 303–319.

Gabrielsson, A., & Lindström, E. 2001. The influence of musical structure on emotional expression. *Pages 223–248 of:* Justin, P. N., & Sloboda, J. A. (eds), *Music and emotion: theory and research*. Oxford University Press.

Geleijnse, G., Schedl, M., & Knees, P. 2007. The quest for groundtruth in musical artist tagging in the social web era. *In: Proc. 8th International Society for Music Information Retrieval Conference*.

Golder, S., & Huberman, B. 2006. Usage Patterns of Collaborative Tagging Systems. *Journal of Information Science*, **32**, 198–208.

Goto, Masataka. 2003. A chorus-section detecting method for musical audio signals. *In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*.

Greenacre, M. J. 1984. *Theory and applications of correspondence analysis*. Academic Press.

Hamel, Philippe, Lemieux, Simon, Bengio, Yoshua, & Eck, Douglas. 2011. Temporal pooling and multiscale learning for automatic annotation and ranking of music audio. *In: Proc. 12th International Society for Music Information Retrieval Conference*.

Hevner, K. 1935. The affective character of major and minor modes in music. *American Journal of Psychology*, **47**, 103–119.

Hevner, K. 1936. Experimental studies of the elements of expression in music. *American Journal of Psychology*, **48**, 246–68.

Hevner, K. 1937. The affective value of pitch and tempo in music. *American Journal of Psychology*, **49**, 621–630.

Hinton, Geoffrey. 2010. *A Practical Guide to Training Restricted Boltzmann Machines*. Tech. rept. UTML TR 2010-003. Department of Computer Science, University of Toronto.

Hofmann, T. 2001. Unsupervised Learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, **42**(1), 177–196.

Hofmann, Thomas. 1999a. Probabilistic Latent Semantic Analysis. *In: Proc. 15th Conference on Uncertainty in Artificial Intelligence*.

Hofmann, Thomas. 1999b. Probabilistic latent semantic indexing. *Pages 50–57 of: Proc. 22nd Annual International ACM SIGIR Conference*.

Hu, Xiao, Downie, J. Stephen, & Ehmann, Andreas F. 2009. Lyric Text Mining in Music Mood Classification. *In: Proc. 10th International Society for Music Information Retrieval Conference*.

Inskip, Charlie, MacFarlane, Andy, & Rafferty, Pauline. 2010. Upbeat and Quirky, With a Bit of a Build: Interpretive Repertoires in Creative Music Search. *In: Proc. 11th International Society for Music Information Retrieval Conference*.

Iwanaga, M. 1997. The affective character of major and minor modes in music. *Perceptual and Motor Skills*, **85**, 287–296.

Jeon, J., Lavrenko, V., & Manmatha, R. 2003. Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. *In: Proc. 26th Annual International ACM SIGIR Conference*.

Kivy, P. 1997. *Philosophies of Arts: An Essay in Differences*. Cambridge University Press.

Knees, P., Pampalk, E., & Widmer, G. 2004. Artist classification with web-based data. *In: Proc. 5th International Society for Music Information Retrieval Conference.*

Knees, Peter. 2004 (November). *Automatische Klassifikation von Musikkünstlern basierend auf Web-Daten*. M.Phil. thesis, Vienna University of Technology.

Knees, Peter, Pohle, Tim, Schedl, Markus, & Widmer, Gerhard. 2007. A Music Search Engine Built upon Audio-based and Web-based Similarity Measures. *In: Proc. 30th Annual International ACM SIGIR Conference.*

Kohavi, Ron, Henne, Randal M., & Sommerfield, Dan. 2007. Practical guide to controlled experiments on the web: listen to your customers not to the hippo. *In: Proc. 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007).*

Kohonen, Teuvo. 1984. *Self-organization and Associative Memory*. Berlin: Springer-Verlag.

Larochelle, Hugo, & Bengio, Yoshua. 2008. Classification using discriminative restricted Boltzmann machines. *In: Proc. 25th International Conference on Machine Learning.*

Larsen, R. J., & Diener, E. 1992. Promises and problems with the circumplex model of emotion. *Pages 25–59 of:* Clark, M. S. (ed), *Emotion: Review of Personality and Social Psychology*. Sage.

Laurier, Cyril, Meyers, Owen, Serr, Joan, Blech, Martn, & Herrera, Perfecto. 2009a. Music mood annotator design and integration. *In: Proc. 7th International Workshop on Content-Based Multimedia Indexing (CBMI 2009).*

Laurier, Cyril, Sordo, Mohamed, Serrà, Joan, & Herrera, Perfecto. 2009b. Music Mood Representations from Social Tags. *In: Proc. 10th International Society for Music Information Retrieval Conference.*

Law, E., West, K., Mandel, M., Bay, M., & Downie, S. 2009. Evaluation of algorithms using games: the case of music tagging. *In: Proc. 10th International Conference on Music Information Retrieval*.

Law, Edith, & von Ahn, Luis. 2009. Input-agreement: a new mechanism for collecting data using human computation games. *In: Proc. 27th Conference on Human Factors in Computing Systems (CHI 2009)*.

Law, Edith, Settles, Burr, & Mitchell, Tom. 2010. Learning to Tag from Open Vocabulary Labels. *Pages 211–226 of:* Balczar, Jos, Bonchi, Francesco, Gionis, Aristides, & Sebag, Michle (eds), *Machine Learning and Knowledge Discovery in Databases*. Lecture Notes in Computer Science, vol. 6322. Springer Berlin / Heidelberg.

Lee, Jin H. 2010. Crowdsourcing Music Similarity Judgments using Mechanical Turk. *In: Proc. 11th International Society for Music Information Retrieval Conference*.

Levy, M., & Sandler, M. 2006a. New methods in structural segmentation of musical audio. *In: Proc. 14th European Signal Processing Conference (EUSIPCO 2006)*.

Levy, M., & Sandler, M. 2007. A semantic space for music derived from social tags. *In: Proc. 8th International Society for Music Information Retrieval Conference*.

Levy, M., & Sandler, M. 2008a. Structural segmentation of musical audio by constrained clustering. *IEEE Trans. Audio, Speech and Language Processing*, **16**(2), 318–326.

Levy, M., & Sandler, M. B. 2008b. Learning latent semantic models for music from social tags. *Journal of New Music Research*, **37**(2), 137–150.

Levy, M., & Sandler, M. B. 2009. Music Information Retrieval Using Social Tags and Audio. *IEEE Trans. Multimedia*, **11**(3), 383–395.

Levy, Mark. 2011. Improving perceptual tempo estimation with crowd-sourced annotations. *In: Proc. 12th International Society for Music Information Retrieval Conference*.

Levy, Mark, & Sandler, Mark. 2006b. Lightweight measures for timbral similarity of musical audio. *In: Proc. 1st ACM Workshop on Audio and Music Computing for Multimedia*.

Levy, Mark, Sandler, Mark, & Casey, Michael. 2006. Extraction of high-level musical structure from audio data and its application to thumbnail generation. *In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*.

Li, T., & Ogihara, M. 2003. Detecting emotion in music. *In: Proc. 4th International Society for Music Information Retrieval Conference*.

Liu, D., Lu, L., & Zhang, H.-J. 2003. Automatic mood detection from music. *In: Proc. 4th International Society for Music Information Retrieval Conference*.

Logan, B., Kositsky, A., & Moreno, P. 2004. Semantic analysis of song lyrics. *In: Proc. IEEE International Conference on Multimedia and Expo (ICME 2004)*.

Lu, L., Wang, M., & Zhang, H. 2004 (October). Repeating Pattern Discovery and Structure Analysis from Acoustic Music Data. *In: 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*.

Maddage, N., Changsheng, X., Kankanhalli, M., & Shao, X. 2004 (October). Content-based Music Structure Analysis with Applications to Music Semantics Understanding. *In: 6th ACM SIGMM International Workshop on Multimedia Information Retrieval*.

Mandel, M., & Ellis, D. 2005. Song-level features and SVMs for music classification. *In: Proc. 6th International Society for Music Information Retrieval Conference*.

Mandel, M., & Ellis, D. 2007. A web-based game for collecting music metadata. *In: Proc. 8th International Society for Music Information Retrieval Conference*.

Mandel, Michael, Poliner, Graham, & Ellis, Daniel. 2006. Support vector machine active learning for music retrieval. *Multimedia Systems*, **12**(1), 3–13.

Mandel, Michael, Pascanu, Razvan, Larochelle, Hugo, & Bengio, Yoshua. 2011a. Autotagging music with conditional restricted Boltzmann machines. *Arxiv preprint arXiv11032832*.

Mandel, Michael I., Eck, Douglas, & Bengio, Yoshua. 2010 (August). Learning tags that vary within a song. *Pages 399–404 of: Proc. 11th International Society for Music Information Retrieval Conference*.

Mandel, Michael I., Pascanu, Razvan, Eck, Douglas, Bengio, Yoshua, Aiello, Luca M., Schifanella, Rossano, & Menczer, Filippo. 2011b. Contextual tag inference. *ACM Transactions on Multimedia Computing, Communications and Applications*. In press.

Manning, C. D., Raghavan, P., & Schütze, H. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Marlin, B., Zemel, R., Roweis, S., & Slaney, M. 2007. Collaborative Filtering and the Missing at Random Assumption. *In: Proc. 23rd Conference on Uncertainty in Artificial Intelligence*.

Marques, G., Domingues, M., Langlois, T., & Gouyon, F. 2011. Three Current Issues in Music Autotagging. *In: Proc. 12th International Society for Music Information Retrieval Conference*.

Mermelstein, P. 1976. Distance Measures for Speech Recognition: Psychological and Instrumental. *Pages 374–388 of:* Chen, C. H. (ed), *Pattern Recognition and Artificial Intelligence*. New York: Academic Press.

Monay, Florent, & Gatica-Perez, Daniel. 2007. Modeling semantic aspects for cross-media image indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**(10), 1802–1817.

Mörchen, F., Ultsch, A., Thies, M., & Löhken, I. 2006. Modelling timbre distances with temporal statistics from polyphonic music. *IEEE Trans. Audio, Speech and Language Processing*, **14**(1), 81–90.

Morgan, R. L., & Heise, D. 1988. Structure of emotions. *Social Psychology Quarterly*, **51**(1), 19–31.

Mori, Y., Takahashi, H., & Oka, R. 1999. Image-to-word transformation based on dividing and vector quantizing images with words. *In: Proc. International Workshop on Multimedia Intelligent Storage and Retrieval Management*.

Oliva, Aude, & Torralba, Antonio. 2001. Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, **42**, 145–175.

Osgood, C. E., Succi, G. J., & Tannenbaum, P. H. 1957. *The measurement of meaning*. University of Illinois Press.

Pachet, F., & Roy, P. 2007. Exploring billions of audio features. *In: Proc. 5th International Workshop on Content-Based Multimedia Indexing (CBMI 2007)*.

Pampalk, E. 2006. *Computational models of music similarity and their application to music information retrieval*. Ph.D. thesis, Vienna University of Technology.

Paulus, J., & Klapuri, A. 2006. Music Structure Analysis by Finding Repeated Parts. *In: Proc. 1st ACM Workshop on Audio and Music Computing for Multimedia*.

Plutchik, R., & Conte, H. R. (eds). 1997. *Circumplex models of personality and emotions*. American Psychological Association.

Popescul, Alexandrin, Ungar, Lyle, Pennock, David, & Lawrence, Steve. 2001. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. *In: Proc. 17th Conference on Uncertainty in Artificial Intelligence*.

Posner, J., Russell, J. A., & Peterson, B. S. 2005. The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, **17**(3), 715–734.

Remington, N. A., Fabrigar, L. R., & Visser, P. S. 2000. Reexamining the Circumplex Model of Affect. *Journal of Personality and Social Psychology*, **79**(2), 286–300.

Rifkin, R., Yeo, G., & Poggio, T. 2003. Regularized Least-Squares Classification. *Pages 131–153 of:* Suykens, J., Horváth, G., Basu, S., Micchelli, C., & Vandewalle, J. (eds), *Advances in learning theory: methods, models and applications*. IOS Press.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. 1986. Learning internal representations by error propagation. *Pages 318–362 of:* Rumelhart, D. E., & McClelland, J. L. (eds), *Parallel distributed processing: explorations in the microstructure of cognition, vol. 1*. Cambridge, MA, USA: MIT Press.

Russell, J. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, **39**(6), 1161–78.

Salton, G., Wong, A., & Yang, C. S. 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, **18**(11), 613–620.

Scaringella, N., Zoia, G., & Mlynek, D. 2006. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, **23**(2), 133–141.

Schifanella, Rossano, Barrat, Alain, Cattuto, Ciro, Markines, Benjamin, & Menczer, Filippo. 2010. Folks in Folksonomies: social link prediction from shared metadata. *In: Proc. 3rd ACM International Conference on Web Search and Data Mining (WSDM 2010)*.

Schlosberg, H. 1941. A scale for the judgment of facial expressions. *Journal of Experimental Psychology*, **29**, 497–510.

Schmitz, P. 2006. Inducing ontology from Flickr tags. *In: Proc. 15th International World Wide Web Conference*.

Schubert, E. 1999. Measuring emotion continuously: validity and reliability of the two-dimensional emotion space. *Australian Journal of Psychology*, **51**, 154–165.

Schubert, E. 2003. Update of the Hevner adjective checklist. *Perceptual and Motor Skills*, **96**, 1117–1122.

Shiu, Y., Jeong, H., & Kuo, C.C. Jay. 2006. Similarity Matrix Processing for Music Structure Analysis. *In: Proc. 1st ACM Workshop on Audio and Music Computing for Multimedia*.

Smola, Alexander J., & Narayanamurthy, Shravan. 2010. An Architecture for Parallel Topic Models. *Proceedings of the Very Large Database Endowment (PVLDB)*, **3**(1), 703–710.

Smolensky, P. 1986. *Information processing in dynamical systems: foundations of harmony theory*. Cambridge, MA, USA: MIT Press. Pages 194–281.

Speck, Jacquelin A., Schmidt, Erik M., Morton, Brandon G., & Kim, Youngmoo E. 2011. A Comparative Study of Collaborative vs. Traditional Musical Mood Annotation. *Pages 549–554 of: Proc. 12th International Society for Music Information Retrieval Conference*.

Stewart, Rebecca, Levy, Mark, & Sandler, Mark. 2008. 3D interactive environment for music collection navigation. *In: Proc. 11th International Conference on Digital Audio Effects*.

Torgerson, W.S. 1958. *Theory and Methods of Scaling*. Wiley.

Turnbull, D., Barrington, L., & Lanckriet, G. 2006. Modeling music and words using a multi-class naive Bayes approach. *In: Proc. 7th International Society for Music Information Retrieval Conference*.

Turnbull, D., Barrington, L., Torres, D., & Lanckriet, G. 2007a. Towards music query-by-semantic-description using the CAL500 data set. *In: Proc. 30th Annual International ACM SIGIR Conference*.

Turnbull, Douglas, Liu, Roy, Barringon, Luke, & Lanckriet, Gert. 2007b. A Game-Based Approach for Collecting Semantic Annotations of Music. *In: Proc. 8th International Society for Music Information Retrieval Conference*.

Turnbull, Douglas, Barrington, Luke, Torres, David, & Lanckriet, Gert. 2008. Semantic Annotation and Retrieval of Music and Sound Effects. *IEEE Trans. Audio, Speech and Language Processing*, **16**(2), 467–476.

Tzanetakis, G., & Cook, P. 2002. Musical genre classification of audio signals. *IEEE Trans. Acoustics, Speech and Signal Processing*, **10**(5).

Vapnik, Vladimir N. 1998. *Statistical Learning Theory*. John Wiley & Sons.

Vesanto, Juha. 2000. Neural Network Tool for Data Mining: SOM Toolbox. *Pages 184–196 of: Proc. Symposium on Tool Environments and Development Methods for Intelligent Systems*. Oulu, Finland: Oulun yliopistopaino.

Vignoli, F., & Pauws, S. 2005. A music retrieval system based on user-driven similarity and its evaluation. *In: Proc. 6th International Society for Music Information Retrieval Conference*.

von Ahn, L., & Dabbish, L. 2004. Labeling images with a computer game. *In: Proc. Conference on Human Factors in Computing Systems (CHI 2004)*.

West, Kris, & Lamere, Paul. 2007. A model-based approach to constructing music similarity functions. *EURASIP Journal on Advances in Signal Processing*, **2007**(1), 149–149.

Weston, Jason, Bengio, Samy, & Hamel, Philippe. 2011. Multi-Tasking with Joint Semantic Spaces for Large-Scale Music Annotation and Retrieval. *Journal of New Music Research*, **40**(4), 337–348.

Whissell, C. M. 1989. The Dictionary of Affect in Language. *Pages 113–131 of:* Plutchik, R., & Kellerman, H. (eds), *Emotion: theory research and experience*, vol. 4. Academic Press.

Whitman, B. 2003. Semantic Rank Reduction of Music Audio. *In: Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*.

Whitman, B. 2005. *Learning the Meaning of Music*. Ph.D. thesis, MIT.

Whitman, B., & Ellis, D. 2004. Automatic record reviews. *In: Proc. 5th International Society for Music Information Retrieval Conference*.

Wieczorkowska, A., Synak, P., Lewis, R., & Raś, W. 2005. Extracting emotions from music data. *In: Proc. 15th International Symposium on Foundations of Intelligent Systems (ISMIS 2005)*.

Wu, X., Zhang, L., & Yu, Y. 2006. Exploring social annotations for the semantic web. *In: Proc. 15th International World Wide Web Conference*.

Yavlinksy, A., Schofield, E., & Rüger, S. 2005. Automated image annotation using global features and robust non-parametric density estimation. *In: Proc. 4th International Conference on Image and Video Retrieval*.

Yoshii, K., Goto, M., Komatani, K., Ogata, T., & Okuno, H. G. Feb. 2008. An Efficient Hybrid Music Recommender System Using an Incrementally Trainable Probabilistic Generative Model. *IEEE Trans. Audio, Speech and Language Processing*, **16**(2), 435–447.